

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

AN AUTOMATED INDIVIDUAL TRAINING RECORD
MANAGEMENT SYSTEM (PROTOTYPE)
UNITED STATES MARINE CORPS

by

Ronald E. Pruiett
David P. Haeusler

March 1984

Thesis Advisor:

Norman R. Lyons

Approved for public release; distribution unlimited

T217479

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Automated Individual Training Record Management System (Prototype), United States Marine Corps		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis March 1984
7. AUTHOR(s) Ronald E. Pruiett David P. Haeusler		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE March 1984
		13. NUMBER OF PAGES 233
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Training, Prototype, Individual System Development, Individual Training Records, Training Management, d Base II		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This thesis examines the feasibility of implementing an existing manual system, the Individual Training Record Management System, on a microcomputer. To demonstrate conceptual feasibility, a prototype is designed and implemented utilizing a commercially available database management system (DBMS). The prototype is not intended to be a fully operational system. Rather it provides an opportunity to demonstrate functions that could be fully implemented in similarly designed systems utilizing the basic criteria of simplicity, utility and low cost.		

Approved for public release; distribution unlimited.

An Automated Individual Training Record Management System
(Prototype),
United States Marine Corps

by

Ronald E. Pruiett
Lieutenant Colonel, United States Marine Corps
E.S., United States Naval Academy, 1967
M.S., University of Southern California, 1982

and

David P. Haeusler
Captain, United States Marine Corps
B.A., University of Colorado, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

ABSTRACT

This thesis examines the feasibility of implementing an existing manual system, the Individual Training Record Management System, on a microcomputer. To demonstrate conceptual feasibility, a prototype is designed and implemented utilizing a commercially available database management system (DBMS). The prototype is not intended to be a fully operational system. Rather it provides an opportunity to demonstrate functions that could be fully implemented in similarly designed systems utilizing the basic criteria of simplicity, utility and low cost.

TABLE OF CONTENTS

I.	INTRODUCTION	11
A.	GENERAL DESCRIPTION	11
E.	BACKGROUND	11
C.	OBJECTIVES	12
D.	ENVIRONMENT	13
E.	SYSTEM CHARACTERISTICS	13
	1. User friendly	13
	2. Reliability	13
	3. Cost	14
	4. Expandability of data base	14
	5. Expandability of functions	14
	6. Speed of operation	14
F.	SUMMARY	14
	1. Scope of Development	14
	2. Concept	15
II.	SYSTEM DESCRIPTION AND REQUIREMENTS	16
A.	INPUT INFORMATION	16
E.	SYSTEM SPECIFICATIONS (FUNCTIONAL)	16
	1. Training Data	16
	2. Data Base Maintenance	17
	3. Report Formats	17
	4. Data Retrieval	17
	5. Standard Reports	18
	6. Decision Support Subsystem	20
	7. Nonstandard Reports	20
	8. System Controls	20
	9. Output Mode	21
C.	REQUIRED SPECIFICATIONS (NON-FUNCTIONAL)	21

I.	DATA STRUCTURE DEFINITIONS (DATA	
	DICTIONARY)	22
	1. Introduction	22
	2. Data Base Management System	22
III.	METHODOLOGY AND DESIGN	24
A.	METHODOLOGY	24
	1. Software	24
B.	SOFTWARE DESIGN	26
	1. Structured Programming	26
	2. Top-down Design	26
C.	DATA FLOW DIAGRAMS	27
D.	SYSTEM HIERARCHY	32
E.	MODULE DESCRIPTIONS	33
	1. ITRMS Master Control Module	33
	2. Access Module	35
	3. Command Determiner Module	36
	4. Nonstandard Report Generator Module	37
	5. Standard Report Generator Module	38
	6. Maintenance Module	39
IV.	IMPLEMENTATION	40
A.	CONSTRAINTS	40
	1. dBASE II	40
	2. Hardware	40
	3. Environmental	41
B.	GENERAL	42
	1. ITRMS Master Control Module	42
	2. Access Module	43
	3. Command Determiner Module	43
	4. Nonstandard Report Generator Module	44
	5. Standard Report Generator Module	46
	6. Maintenance Module	56

V.	CONCLUSION	64
A.	GENERAL	64
E.	SYSTEM SPECIFICATIONS (FUNCTIONAL)	65
	1. Storage Limitations	65
	2. Retrieval Limitations	65
C.	SYSTEM SPECIFICATIONS (NON-FUNCTIONAL)	65
	1. Training	65
	2. Error Checking	66
	3. Table Mapping	66
D.	SUBSEQUENT SYSTEM DEVELOPMENT	66
E.	SUMMARY	67
	APPENDIX A: DATA ELEMENT DICTIONARY	68
A.	FIELD	68
E.	NAME	68
C.	TYPE	68
D.	WIDTH	69
E.	DESCRIPTION	69
F.	DATA BASE FILES	69
	1. Pers (Personnel)	69
	2. EST (Essential Subjects Testing)	70
	3. QUAL (Qualifications)	70
	4. INFOMISC (Information Miscellaneous)	71
	5. SECURITY (Access Information)	71
	6. COMMENTS (Non-formatted Information)	71
	APPENDIX E: DECISION SUPPORT SYSTEM (DSS)	
	DESCRIPTION	73
A.	BACKGROUND	73
E.	PURPOSE	73
C.	FUNCTIONAL DESCRIPTION	73
D.	USER'S VIEW	75
	1. Summary Information	75
	2. Support Depth	76

3. Interdependence	76
4. Restrictions	76
E. BUILDER'S VIEW	76
1. Dialogue	76
2. Modelling	76
F. CONTENTS CF KNOWLEDGE BASE	77
APPENDIX C: ITRS MASTER CONTROL MODULE LISTING	78
A. MASTER ROUTINE	78
E. PAUSE ROUTINE	80
APPENDIX D: ACCESS MCDULE LISTING	81
A. ACCESS ROUTINE	81
APPENDIX E: COMMAND DETERMINER MODULE LISTING	83
A. COMMAND DETERMINER ROUTINE	83
E. HELP2 ROUTINE	84
APPENDIX F: NONSTANDARD COMMAND GENERATOR MOUDLE	
LISTING	86
A. NONSTANDARD REPORT ROUTINE	86
E. NONSTANDARD HELP ROUTINE	99
C. SELECT ATTRIBUTE ROUTINE	100
D. SELECT DESCRIPTOR ROUTINE	104
APPENDIX G: STANDARD COMMAND GENERATOR MODULE	
LISTINGS	109
A. STANDARD REPORT ROUTINE	109
E. SELECT UNIT ROUTINE	111
C. UNIT ROSTER ROUTINE	113
D. TRAINING STATISTICS ROUTINE	115
E. TRAINING CCUNT ROUTINE	126
F. QUALIFICATION COUNT ROUTINE	128
G. TRAINING FCSTER ROUTINE	129
H. TRAINING FCSTER FORMAT ROUTINE	132
I. MARKSMANSHIP STATUS ROUTINE	133

J.	CONVERT UNIT ROUTINE	135
K.	QUOTA STUDY ROUTINE	136
L.	QUOTA SCRATCH PAD ROUTINE	140
M.	MARKSMANSHIP ROSTER ROUTINE	155
N.	MOS STATUS ROUTINE	157
O.	MOS ROSTER ROUTINE	159
P.	PERSONNEL DATA ROUTINE	161
Q.	PERSONNEL DATA FORMAT ROUTINE	164
R.	EST STATUS ROUTINE	165
S.	EST COUNT ROUTINE	170
T.	EST ROSTER ROUTINE	171
U.	EST REPORT FORMAT ROUTINE	174
V.	SWIM QUALIFICATION ROUTINE	175
W.	SWIM REPORT ROUTINE	178
X.	HELP ROUTINE	179
APPENDIX H: MAINTENANCE MODULE LISTING		186
A.	MAINTAIN CONTROL ROUTINE	186
B.	VIEW CONTROL ROUTINE	188
C.	ITR SCREEN FORMAT ROUTINE	190
D.	SCREEN DATA RETRIEVAL ROUTINE	192
E.	COMMENT RETRIEVAL ROUTINE	194
F.	UPDATE CONTROL ROUTINE	195
G.	DATA UPDATE ROUTINE	197
H.	ITR CREATION CONTROL ROUTINE	206
I.	ITR DATA INITIALIZATION ROUTINE	207
J.	DATA ERROR CHECKING ROUTINE	211
K.	CALCULATION ROUTINE	214
L.	COMMENT UPDATE ROUTINE	219
M.	COMMENT CREATION ROUTINE	220
N.	DELETE ITR ROUTINE	221
O.	SYSTEM FUNCTION CONTROL ROUTINE	223
P.	SYSTEM INSTALLATION ROUTINE	224
Q.	SYSTEM RESET ROUTINE	225

R. SYSTEM ACCESS CONTROL ROUTINE	229
BIBLIOGRAPHY	232
INITIAL DISTRIBUTION LIST	233

LIST OF FIGURES

3.1	Initial Data Flow Diagram	28
3.2	Data Flow Diagram Expansion for DBMS	29
3.3	Data Flow Diagram Expansion 2	30
3.4	Data Flow Expansion for Access Function	31
3.5	Data Flow Expansion for Command Generators	32
3.6	ITRM System Hierarchy	34
4.1	Command Determiner Menu	44
4.2	Attributes and Descriptors Display	45
4.3	Standard Report Menu	47
4.4	Personnel Roster Report	48
4.5	Training Status Report	49
4.6	Marksmanship Status Report	51
4.7	Marksmanship Roster Report	53
4.8	MOS Status Report	53
4.9	MOS Roster Report	54
4.10	EST Status Report	55
4.11	Swimming Qualification Status Report	56
4.12	Maintenance Module Menu	57
4.13	ITR Screen Format	58
4.14	System Function Menu	62
4.15	Access Function Menu	63
B.1	Memory File	77

I. INTRODUCTION

A. GENERAL DESCRIPTION

This interactive prototype automates the existing manual system of maintaining individual training information on USMC personnel. The term "system" is used throughout this thesis to mean hardware and software. The software consists of customized code which overlays a commercially available data base management system (DBMS). The system is targeted for implementation at the infantry or artillery battalion or subordinate unit level. However, with minor modifications, primarily increasing the limitation on the number of subordinate units and increasing the size of the database, the system could be used by any U.S.M.C. organization, including the Marine Air Ground Task Force (MAGTF). No prior user experience in computer or automated systems is assumed. In addition to generating standard reports, the system provides limited decision support capabilities, responds to non-standard data base inquiries, and includes facilities for data base maintenance.

E. BACKGROUND

Individual training subjects in the U.S. Marine Corps encompass a variety of otherwise unrelated training topics that are conducted and evaluated periodically. The common characteristic of individual training vis-a-vis unit training is that training and testing relate to individual skills and individual levels of performance. Since the focus is on individual training and individual testing, data is maintained for each individual in a unit and consequently, the system is very data intensive.

Individual training includes subjects such as "essential subjects", physical fitness, rifle/pistol qualifications, career training, and military occupational specialty (MOS) training. Each subject or grouping of subjects typically includes its own set of training and testing requirements. For example, some individual training subjects require proficiency demonstration only once during an entire career while others must be evaluated annually. In total individual training information includes approximately 60 separate elements of information that must be maintained for each Marine. Individual Training Records (ITR's) are traditionally maintained manually at the company level. A data base consisting of 200 records and 10,000 data elements in a company size unit is not uncommon. In addition to data base maintenance, which includes frequent creation, deletion, and update of records, the data base is often used to provide training management information concerning the current training status of the unit or as a basis for developing future training plans.

It is the contention of this thesis that the system can be operated and maintained by a technically unsophisticated user at a reasonable cost.

C. OBJECTIVES

The objective of this thesis is to design and implement a prototype system that is simple to use and that:

1. Maintains individual training records for a USMC battalion;
2. Provides standard reports upon request;
3. Provides nonstandard reports upon request;
4. Demonstrates a decision support application;
5. Provides a data base maintenance capability.

D. ENVIRONMENT

The system is used and maintained by persons with no prior experience with computers and with little prior training on this system. The system's data base requires frequent maintenance. Approximately a 30% record turnover rate (i.e., 30% of old records will be deleted and 30% new records will be inserted into the data base) is expected each year. Additionally, records will be modified at the rate of approximately 50 data fields per work day for each company (based on an average strength of 150 Marines per company). The physical environment envisioned is a typical office setting with no unique system requirements anticipated.

E. SYSTEM CHARACTERISTICS

The following system characteristics are listed in order of importance.

1. User friendly

The system is simple to operate and maintain. It requires no extended training sessions and it provides prompts to the user.

2. Reliability

The reliability of the hardware and the DBMS is beyond the scope of this thesis. The code written to implement this design is the user-system interface. Accordingly, to ensure the integrity of the system, the DBMS remains transparent and inaccessible to the user. The overall reliability of the system is high. To this end, the system includes the following provisions:

a. Menu driven

b. No possibility of accidental erasure of data base.

c. No possibility of user to "escape" or default to DBMS.

3. Cost

The costs of a microcomputer and the proprietary software required to run the system are low compared to mainframe or mini-computer application.

4. Expandability of data base

The data base can be expanded subject to memory and DBMS constraints. The larger the data base the slower the system will respond to user requirements. This is, however, an acceptable tradeoff.

5. Expandability of functions

The system can be easily expanded to include added functions. This facility has been implemented by incorporating a modular structure in the design. A module can be added or modified with little difficulty.

6. Speed of operation

The speed of the system is relative to the size of the data base. However in all cases compared to the present manual system it is much faster.

F. SUMMARY

1. Scope of Development

This thesis includes the determination of system requirements, system feasibility, system design and review, and implementation of a functional prototype.

2. Concept

The purpose of this thesis is to demonstrate feasibility of the concept: A fully automated individual training record management system for a battalion size unit can be implemented on a microcomputer. The system can be developed and operated at a reasonable cost. Furthermore, the system can be operated and maintained by persons having no prior computer experience after little training.

II. SYSTEM DESCRIPTION AND REQUIREMENTS

A. INPUT INFORMATION

The user's input consists of a sequence of menu prompted responses. Each user's response is either a specific request to the system for data base maintenance, a standard report, or data necessary to generate the desired nonstandard report.

B. SYSTEM SPECIFICATIONS (FUNCTIONAL)

1. Training Data

The system is capable of developing and maintaining training data for at least 800 Marines. The following information is maintained for each individual:

- a. Name/SSN/personal data
- b. Rank
- c. Unit (company and platoon)
- d. Annual training requirements status (essential subjects, leadership, SNCO/NCO training ,etc.)
- e. Rifle/pistol qualifications status (current year)
- f. MOS qualifications
- g. An unformatted data field to be utilized as desired by the user. It may be used to maintain the following kinds of information:
 - (1) Formal schools attended.
 - (2) Local schools attended.

(3) MCI courses completed.

(4) Participation in training exercises/
training deployments.

2. Data Base Maintenance

Maintenance of the data base is simple and provides positive user control at all times (i.e., user inputs data and views updated individual training record before data is stored into memory).

3. Report Formats

Most standard reports are retrievable in a statistical mode or in a unit roster mode.

a. Statistical Mode

The statistical mode report provides the total or the percent of a unit that has satisfied a particular user specified training criterion (What percent of a unit is swim qualified?).

b. Roster Mode

The roster mode report provides names of persons in a specified unit that have satisfied a training criterion.

4. Data Retrieval

All standard reports are retrievable by unit designator (e.g., battalion, company/platoon). Training data is retrievable by individual, by unit, or by a user defined set of attributes (e.g., all sgts in A CO) utilizing the nonstandard reports option.

5. Standard Reports

The system provides the following standard reports:

a. Unit Roster

The Unit Roster Report will display in alphabetical order the NAME, RANK, and PRIMARY MOS for each member of the selected unit or subunit.

b. Personal Data Report

The Personal Data Report displays for each member of a unit or subunit all personal data including NAME, SSN, RANK, PRIMARY/SECONDARY MOS, UNIT/SUBUNIT NAME, JOIN DATE, EAS, BIRTHDATE, HEIGHT, WEIGHT and a one character boolean comment block.

c. Essential Subject Training Requirements Status Report

The Essential Subject Training Requirements Status Report displays the percent and total number of unit or subunit members that have completed an Essential Subject requirement or the PFT (physical fitness test).

d. Annual Training Requirements Status and Roster Reports

The Training Status Report displays for each essential subject, the percent of the unit that has successfully completed each training element and the total number of unit members that have completed each training element. In addition, the Training Status Report includes completion statistics for PFT1, PFT2, SWIMQUAL, RIFLE QUAL, and PISTOL QUAL.

The Training Roster Report displays in alphabetical order each member of the selected unit or subunit and

his complete training status for the current year. If a training element has been successfully completed for the current year, a letter corresponding to that training element will be shown. An "*" symbol indicates that a training element has not been completed. For example, "FIRST AID AE*H**B**DF" indicates that elements ABDEF and H have been completed and five of the eleven elements have not been completed (i.e., C,G,I,J and K).

e. Rifle/Pistol Qualifications Status

and Roster Reports

The Marksmanship Status Report will display the qualification results for the unit or subunit members that have fired the rifle or pistol for qualification during the current year. Also, the percent that have qualified in each category (eg. EX,SS,MM,UN) and current year results for rifle and pistol will be shown. The pistol qualification results include only those members required to fire the pistol for qualification.

The Marksmanship Roster Report will display in alphabetical order the NAME, RANK, DATE OF QUALIFICATION, and QUALIFICATION RESULTS for each member of the selected unit or subunit. Current year results only will be displayed.

f. MOS Qualification Status and Roster Reports

The MOS Roster Report groups the selected unit or subunit by primary MOS and RANK and then displays in alphabetical order the NAME, RANK, PRIMARY MOS AND SECONDARY MOS of all members in the unit or subunit. This report will function regardless of the number or different types of MOS's contained in a unit.

g. Individual Training Record Report

The Individual Training Record Report prompts the user for the name of a member in the data base. After a name is entered the complete individual training record for that individual is displayed. This includes all training data included in the data base on this particular individual (i.e., essential subject status, PFT status, current year rifle/pistol qualification results, swim qualification status) and personal data (i.e., name, rank, SSN, unit, primary MOS).

h. Swim Qualification Report

The Swim Qualification Report displays in alphabetical order for each member of the unit or subunit selected NAME, RANK, UNIT and current SWIM QUAL results.

6. Decision Support Subsystem

The purpose of this component is simply to demonstrate the decision support potential of this system using the available data base.

7. Nonstandard Reports

The system generates user defined non-standard reports. These reports are created from attributes that the user determines. A maximum of three attributes can be selected.

8. System Controls

Safety controls are provided to ensure that the data base cannot be accidentally written over or otherwise destroyed. Also, the system includes an executive access control facility, and control for "read only" access or "read/write" access. System access is permitted in one of the following three levels:

a. Data Base Administrator level is intended for the sole use of the Data Base Administrator and includes full system capabilities (install system, initialize data base, modify access directory, create/ delete/modify data base records).

b. Create/modify data base level

c. Restricted to retrieve information in report format only from the data base.

9. Output Mode

The prototype is capable of sending reports to a CRT screen. With minor modifications the user can be given an option of screen or printed copy output.

C. REQUIRED SPECIFICATIONS (NON-FUNCTIONAL)

The following is a list of non-functional system specifications:

1. The system is interactive.
2. A high school graduate with no computer background can retrieve standard reports with no more than one hour of system training.
3. A high school graduate with no computer background can create and maintain the data base with no more than six hours of training.
4. A high school graduate with no computer background can retrieve a standard report after one hour of training.
5. A high school graduate with no computer background can retrieve a multiple-criteria nonstandard report after an additional one hour of training.
6. The system can be easily modified to change record formats, field formats, and standard report formats

without a major redesign or reprogramming effort. A major redesign is considered any design effort that involves changing a module interface.

7. The system can retrieve standard reports in 15 minutes or less.
8. The system can retrieve non-standard reports in 30 minutes or less.
9. The system does "out of bounds" checking/parameter checking on all quantitative data and standard inputs.
10. The system automatically converts rifle, pistol, and PFT raw scores into corresponding qualifications (e.g. 245 rifle raw score ==> Expert).

D. DATA STRUCTURE DEFINITIONS (DATA DICTIONARY)

1. Introduction

The data dictionary should be maintained by the Data Base Administrator and documentation should be accessible to all users. This data dictionary facilitates maintenance, the integrity of the information in the data base and provides information on data relationships.

2. Data Base Management System

a. Commercial Software

The Data Base Management System (DBMS) kernel is a commercially available software system. The program dBase II, a relational DBMS, provides all of the necessary attributes (availability, record capacity, user friendly environment, etc.). Therefore, dBase II (Ashton-Tate version 2.4) is used, and data structures are implemented by the dBase II system.

k. Data Base Files

The information maintained on each individual is organized in a relational data base. The following records or tuples are defined for each individual (The number of characters per field is defined in the data element dictionary, see Appendix A) :

(1) Personal Information Record. SSN, name, military occupational specialty (MOS), rank, weight, height, company/platoon, birthdate, joindate, expiration of active service (EAS), and gas mask size.

(2) Essential Subject Training Requirements. SSN, 9 essential subjects, and physical fitness information.

(3) Rifle/Pistol/Swimming Qualifications. SSN, rifle score current year, pistol score current year, swimming qualifications, and dates.

(4) Information Miscellaneous. SSN, 11 fields for training information.

(5) Comment. SSN; 250 characters that can be utilized, at the users discretion, for formal/informal school data, deployments information, MCI courses completed or specific comments.

(6) Security. SSN, name, user identification code, password, and authorization level.

III. METHODOLOGY AND DESIGN

A. METHODOLOGY

1. Software

At the center of the software subsystem is the IBM Personal Computer Disk Operating System (DOS 2.0). System requirements for PC DCS operating systems includes:

- a. 8088 based microprocessor
- b. 128K bytes of RAM memory
- c. Cursor addressable 24 line by 80 column CRT

The software layer immediately outside DOS 2.0 is dBase II - (version 2.4) which is a commercially available relational data base application system developed and distributed by ASHTON-TATE. dBase II is a data base management tool that permits manipulation of user designed data base files using English-like commands which collectively define a query language.

The layer of code contiguous with and completely surrounding dBase II is what we designed and wrote to satisfy the thesis objective. The code is written in the dBase II query language. This layer of code, that will be referred to as the "cutter layer", is designed to be the only interface between the system user and the system. All system entries, data base transactions or manipulations, report inquiries, and system exits are through the cutter layer of code. This is an important and necessary design consideration for the following reasons:

a. Simplicity

In order to satisfy the system requirement for simplicity, it was necessary to impose a software layer between the user and the dBase II software. This outer layer presents to the user a series of menus. The user responds to each menu prompt by pressing the keyboard key that corresponds to action desired. The user's response in turn initiates a code sequence that causes the desired action to be taken. If additional user input is required to complete the action, the code causes another menu to be presented to the user. Each menu represents an exhaustive list of possible options that may be selected by the user. This sequence of menu presentation and user response is repeated until the desired action has been adequately defined. At this point the code executes using as variables the input provided by the user. Therefore, this outer layer includes all of the code that determines the functions available to the user and simplifies the operation of the system.

b. Layering

If the user had access to the DOS 2.0 or the dBase II layer he could easily modify the outer layer code. This action could cause a system failure. Therefore, this eventuality must be prevented. Another function of the outer layer is to prevent access to inner layers of software.

c. Data Integrity

Since the data base files contain personal data, that needs to be protected, it is important to limit access to the data base. Also, uncontrolled access to data base files can result in lost data or catastrophic destruction of

the data base. To preclude this, the outer layer provides system security and ensures data integrity.

d. File Maintenance

Whenever data base files are accessed, house-keeping functions need to be performed such as opening or closing files, joining files, creating temporary files or deleting temporary files when no longer needed, and creating or releasing variables. The outer layer automatically performs these functions for the user.

E. SOFTWARE DESIGN

1. Structured Programming

Although the dBase II query language is not a structured programming language, structured programming techniques were employed during the software design. This was done to reduce system maintenance, to simplify system design and to make future changes to the system easier to implement. The following techniques were emphasized during the design phase of the development cycle.

- a. Top down design and hierarchical structure
- b. Modularization
- c. Information hiding
- d. In general, efforts were made to maximize module cohesion and to minimize coupling between modules.

2. Top-down Design

The overall design concept was the top-down design or step-wise refinement technique. This concept provides the capability to work from a simple idea and expand it to the final complex product. This process facilitates orderly

and logical development of the design steps. The first step in the process is to determine how the data or an input is transformed to produce the desired output. The concentration is on 'What' is done not 'How' it is done. With this information data flow diagrams are produced. These diagrams should be considered arrows pointing to 'black boxes' which manipulate the data. The actual mechanisms of the manipulations are not important at this point in the design process. The diagrams show the information passing through the system and the transformation of data in the system. These diagrams are produced at an increasing level of detail until further expansion does not provide a significant change.

From the data flow diagrams, hierarchical charts are produced. Any information modifications that are related, are grouped into the same module. Once the hierarchy is established, algorithms are developed using structured programming techniques. These algorithms are then converted into code. The code is implemented using the top down programming technique, this technique implements the highest levels of the hierarchy first. As lower level modules are completed and tested, they are added to the existing modules. This technique simplifies a complicated program.

C. DATA FLOW DIAGRAMS

At first, it is necessary to decide what information goes into the system and what information exits from the system. The user inputs both data and commands from a terminal and receives, from the system, reports and information. The first iteration of the data flow diagram has two inputs and two outputs (see Figure 3.1).

The Individual Training Record Management (ITRM) System is responsible for maintaining a large amount of data. Handling of the information should be transparent to the

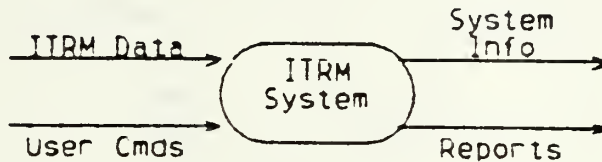


Figure 3.1 Initial Data Flow Diagram

user. Therefore, a Data Base Management System (DBMS) was incorporated to maintain the data base. The DBMS requires specific commands as input and outputs information (see Figure 3.2).

The ITRM System is designed to have a restricted access. The user must enter the proper identification for the system to allow access. The access function produces a valid/invalid acknowledgement to the remaining portion of the ITRM system. Commands and data pass through the Command Determiner which selects the type of action the system is to execute. The three functions called by the Command Determiner are:

1. Standard Command Determiner
2. Nonstandard Command Determiner
3. Maintenance Handler

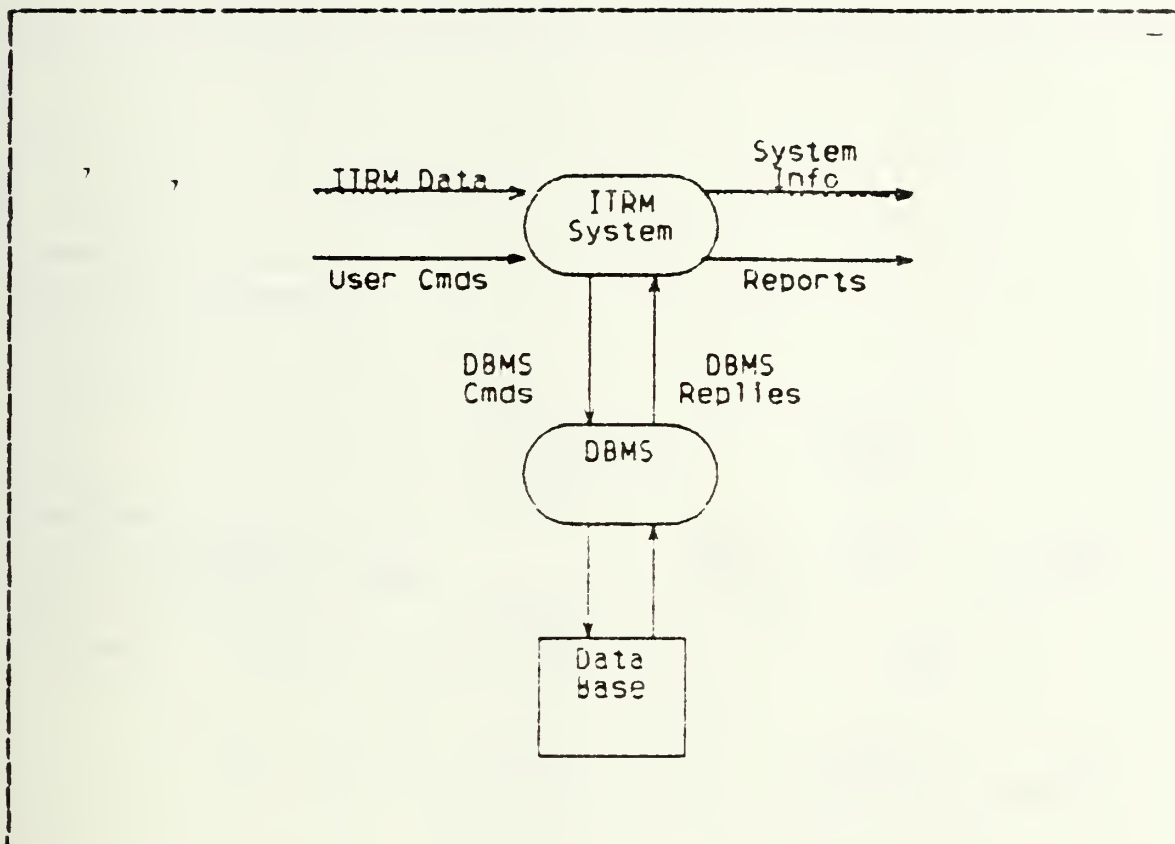


Figure 3.2 Data Flow Diagram Expansion for DBMS

These three functions interface with the DBMS by producing DBMS commands. The DBMS then communicates to the Report Generator the information necessary to produce the desired reports. Figure 3.3 shows the second expansion.

The access control of any system can be very complex. This system requires the user to input an identification code and a password. The system then matches these items to insure an authorized user. If access is valid, the user's authorization level is passed to the rest of the system. If access is invalid, the user gets one more opportunity to enter a correct access sequence. If the user fails again the system terminates the session. The information required to handle the access function is stored in the DBMS. The

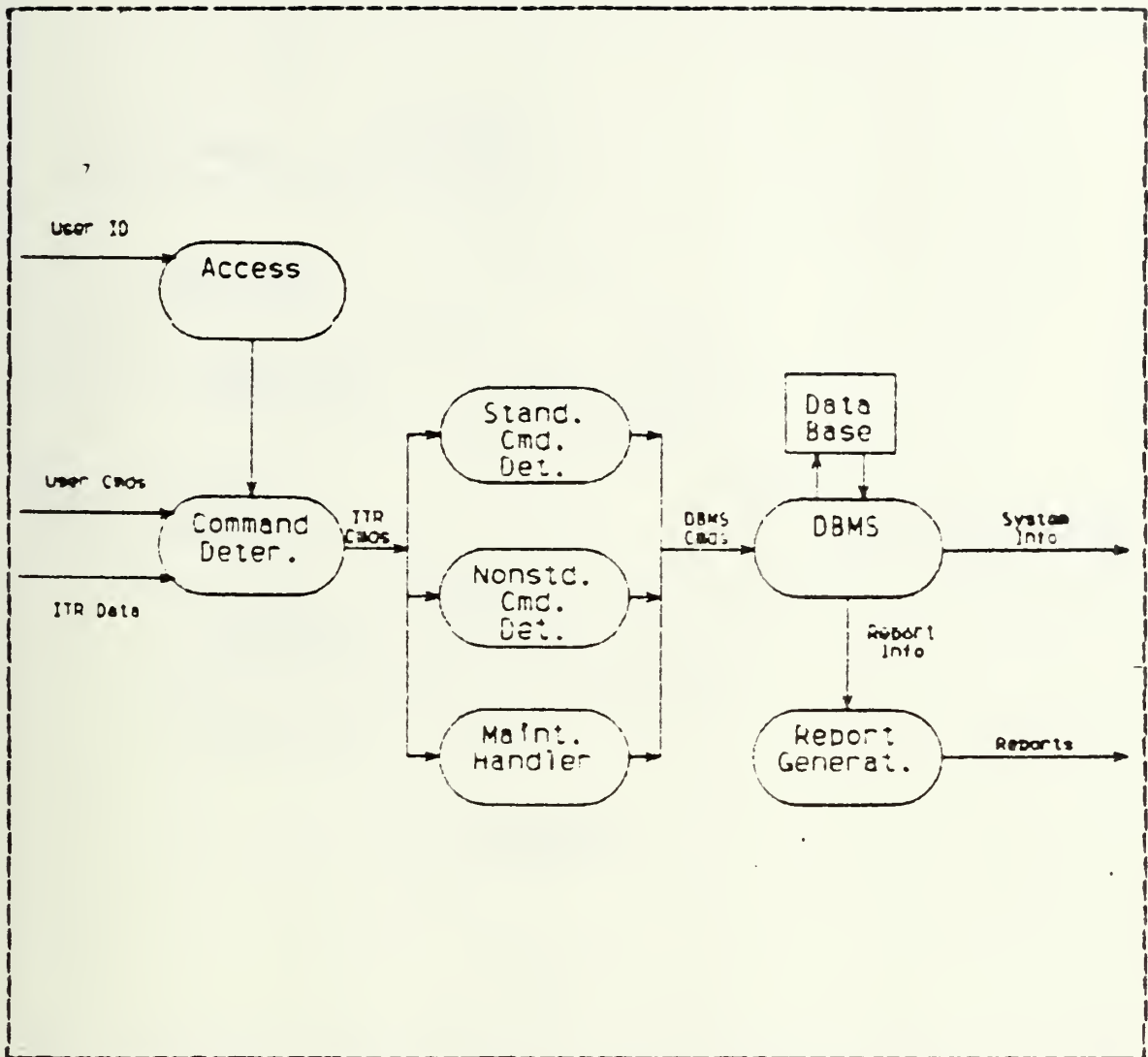


Figure 3.3 Data Flow Diagram Expansion 2

premise of the entire system is to prompt the user into entering the correct response, thereby reducing the need for any prior experience. The ITFM system prompts the user for the proper access codes and informs him when an error has been made. Figure 3.4 displays the Access data flow expansion.

Each of the Command Determiners prompts the user for responses from which DBMS commands are derived. The report

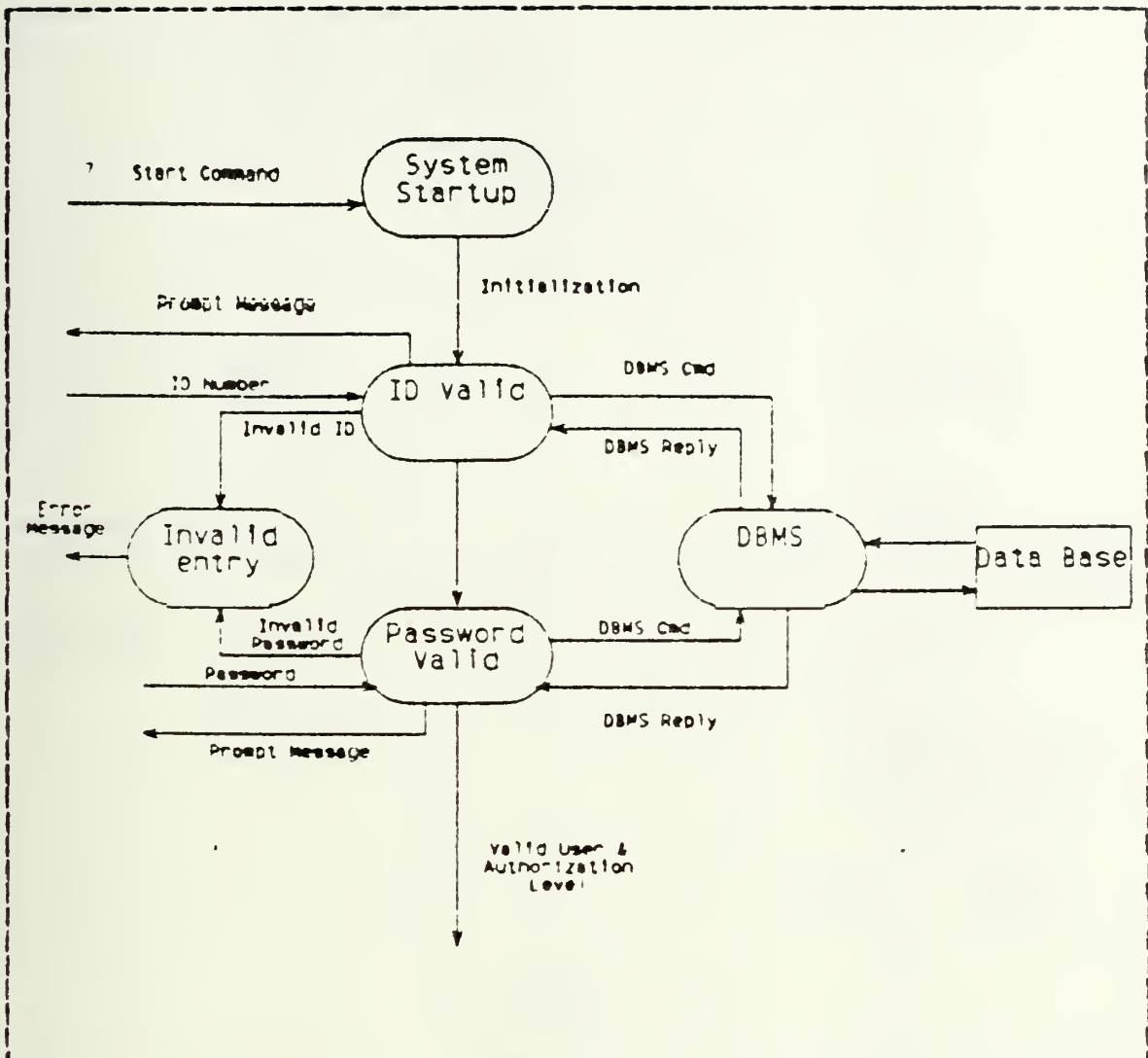


Figure 3.4 Data Flow Expansion for Access Function

generator then interprets the DBMS replies and prompts the user to format the reports (Figure 3.5).

The data flow diagrams, demonstrate the simple and logical flow of the data through the ITRM System. The next step is to logically organize the flow diagrams into a hierarchy of modules that accomplish the data transformations depicted in the data flow diagrams.

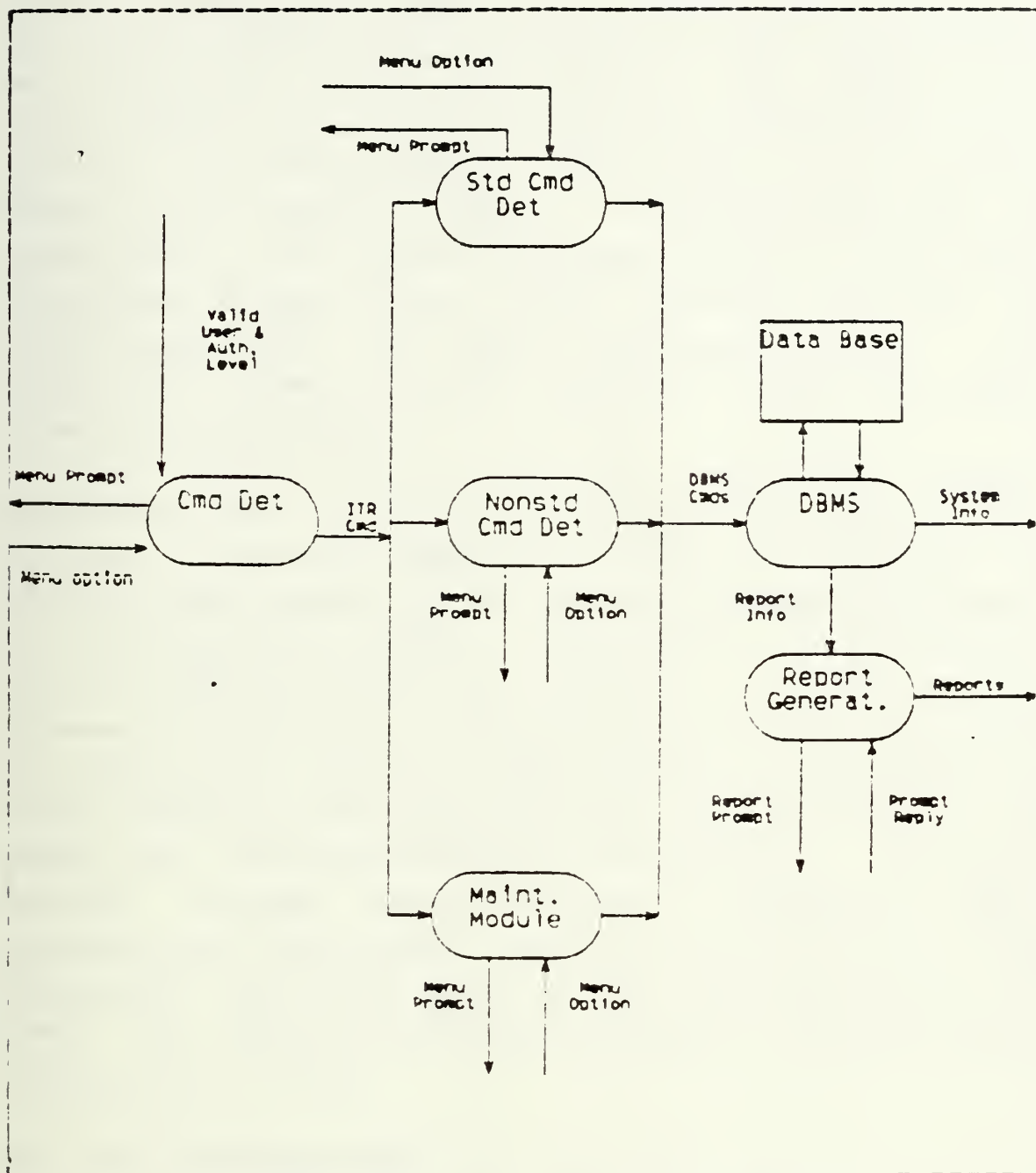


Figure 3.5 Data Flow Expansion for Command Generators

D. SYSTEM HIERARCHY

The ITRM System is relatively simple in design. It requires maintenance of the data base, production of

standard reports, and simple query capabilities for the user through nonstandard reports. The system must be initialized and sequential control established. The ITRMS Master Control Module was created to handle the functions of system control and initialization. This module is the highest level of the hierarchy, level 1. All access functions are consolidated in one module, the Access Module. The Command Determiner Module controls which functions the user can invoke and checks for proper authorization level. The Access and Command Determiner Modules comprise the second level of the hierarchy.

The Standard Command Determiner, Nonstandard Command Determiner, and the Maintenance Handler are combined with portions of the Report Generator to form three modules, the Standard Report Generator, the Nonstandard Report Generator, and the Maintenance Module. These three modules interface with the DEMS and produce DBMS commands. In addition, they process the information returned from the DBMS and produce the appropriate reports. Consolidation of determiner and report generation functions simplifies the structure of the system and facilitates subsequent maintenance. The three modules, Standard Report Generator, Nonstandard Report Generator, and Maintenance, comprise the third level of the hierarchy. The lowest layer of the hierarchy contains the DBMS. The hierarchy displayed in Figure 3.6 demonstrates simplicity of design.

E. MODULE DESCRIPTIONS

1. ITRMS Master Control Module

a. Module Purpose

The ITRMS Master Control Module initializes the system and controls the logical sequencing of all modules.

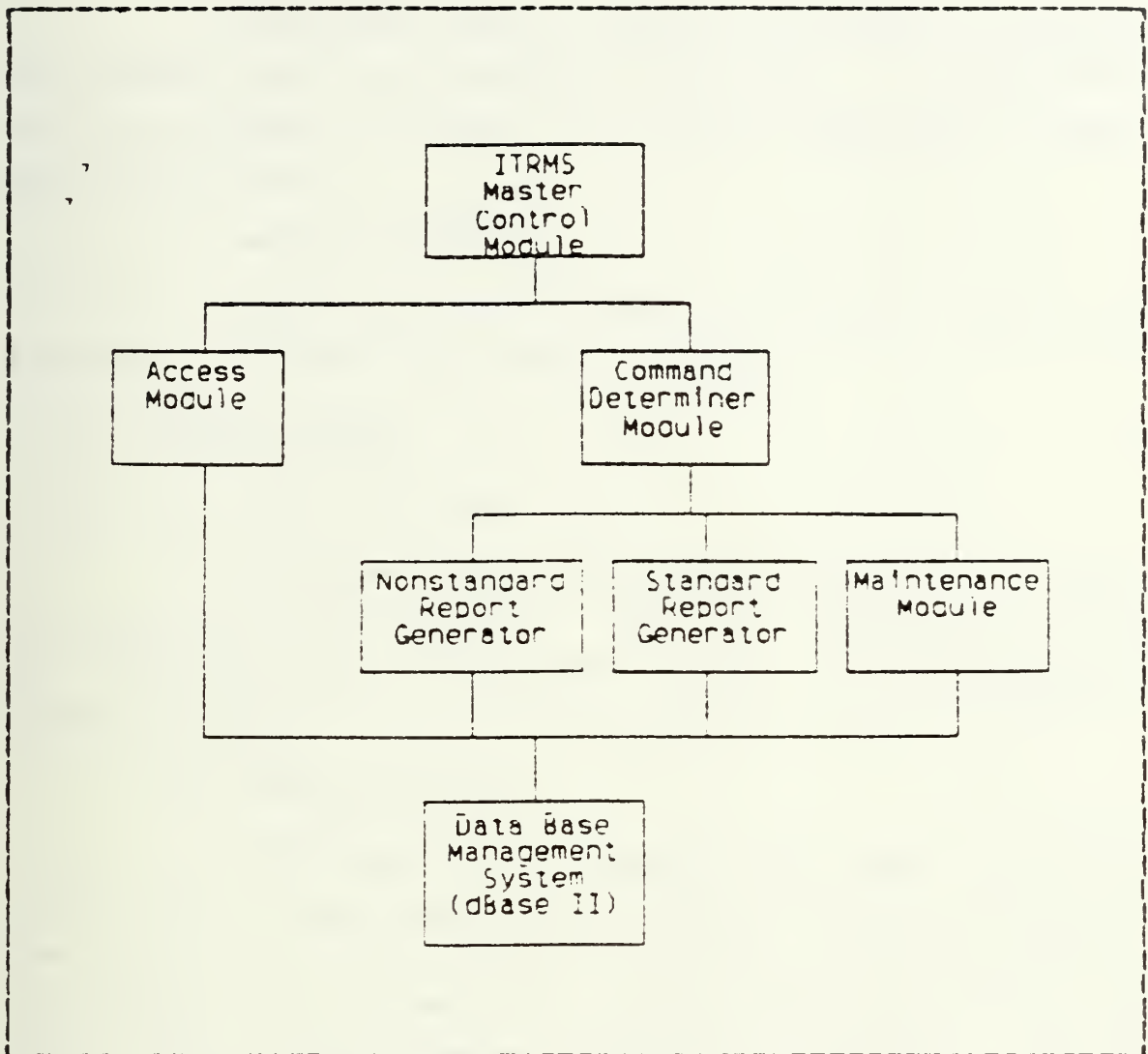


Figure 3.6 ITRM System Hierarchy

1. Module Function

The ITRMS Master Control Module initiates and coordinates all internal system processes (except operating system functions). The module responds to internal cues (from other modules) and to environmental cues (from the user). The module controls the access to all modules and shields the user from the resident operating system. The module will not allow the user to exit to the operating system.

c. Interfaces

The master control module is at level 1 which is the highest level in the module hierarchy. It interfaces with two modules at level 2, Access Module and Command Determiner Module.

d. Design Decisions

This module can be easily modified to handle any expansions by adding new module interfaces.

2. Access Module

a. Module Purpose

The Access Module prompts the user for an identification code and password. The module determines if the user has authorized access and assigns the corresponding level of access.

b. Module Function

The Access Module requests the user's identification code and password. It validates the ID to ensure the user is on the access list and to ensure that the password corresponds with the user ID. If the user ID is not on the access list an appropriate message is returned. If the password does not correspond with the user ID (after two iterations), an appropriate message is displayed and the session is terminated. Once the correspondence of ID and password has been determined, the module passes the access authorization level to the command determiner. The authorization level is used to determine which menu options are available to the specific user.

c. Interface

The Access Module receives control instructions from the ITRMS Master Control Module. The Access Module interfaces directly with the DBMS to manipulate the security file. After the access module completes its processing and if the user is authorized access, the level of access is passed back to the ITRMS Master Control Module.

d. Design Decision

The design allows modifications to be made to the access system without side effects.

3. Command Determiner Module

a. Module Purpose

The Command Determiner Module directs control to either the Standard Report Generator, Nonstandard Report Generator, or the Maintenance Module.

b. Module Function

The Command Determiner asks the user, through a standard set of menu prompts, what task he wishes to accomplish. Based on the user response, the Command Determiner Module transfers control to one of the three lower level modules. The Command Determiner Module also checks to see if the user has access to that module. If access is not authorized then an appropriate message is displayed.

c. Interfaces

The Command Determiner is called by the ITRMS Master Control Module and passes control to either the Standard Report Generator, Nonstandard Report Generator, or the Maintenance Module.

c. Design Decisions

This module allows the separation of the Standard Report Generator, the Nonstandard Report Generator, and the Maintenance Module. This separation facilitates changes and reduces module coupling.

4. Nonstandard Report Generator Module

a. Module Purpose

The Nonstandard Report Generator allows the user to query the data base in a controlled manner. The queries are generated from a series of prompts to the user.

b. Module Function

The module prompts the user, through a series of menus, to produce the desired query. When the module receives the appropriate prompted input from the user, the nonstandard command generator accesses the DBMS for the requested information. The information is then formatted and displayed.

c. Interfaces

The Command Determiner Module calls this module based on user selection. The Nonstandard Report Generator interacts directly with the user. This module constructs the queries and communicates them to the DBMS. It also processes the information received from the DBMS and formats the data into a report.

d. Design Decisions

If there is a future need to modify the types of queries that are available to the user, the changes will be confined to this module.

5. Standard Report Generator Module

a. Module Purpose

The Standard Report Generator produces the DBMS commands that will create the standard reports. These reports are produced on a regular basis, therefore, it is advantageous to standardize the reports.

b. Module Function

The Standard Report Generator displays option menus to the user and translates user's responses into the necessary sequence of DBMS instructions. After the user decides which standard reports will fulfill his needs, the module formats standard reports and displays the user requested information. The Standard Report Generator includes a CSS subset.

c. Interfaces

The Standard Report Generator is called by the Command Determiner. The Standard Report Generator communicates with the DBMS and processes information from the DBMS to produce its reports.

d. Design Decisions

This module was designed to take advantage of the fact that many applications are routine in nature. These are programmed in a single module and significantly simplify access to routine reports. If any of these routine reports need modification, the change can be isolated to this module.

6. Maintenance Module

a. Module Purpose

The Maintenance Module provides users the capability to create, delete, and update the data base records. Necessary system functions reside in this module. The Data Base Administrator maintains the access file from this module and initializes the system to his specific unit.

b. Module Function

To update a record, the Maintenance Module extracts the data in each data base file for a selected unit member. The data is combined into a single screen display representing all data contained in the system on the individual. To update a field, the data in the field can be typed over with current data. Upon completion, each data element is read and returned to its proper location in the appropriate data base file. Creation of new records is accomplished in the same manner. Records are deleted by a single deletion command. Therefore, this command usage is limited to avoid accidental and malicious destruction of data.

c. Interfaces

The Maintenance Module interfaces with the Command Determiner and the DBMS.

d. Design Decisions

All data modifications are implemented using the Maintenance Module. The physical and logical structure of all data base files is hidden from the user.

IV. IMPLEMENTATION

A. CCNSTRRAINTS

1. dBASE II

Following is a list of dBase II constraints that affected the design of this system and consequently imposed constraints on the implementation of the Individual Training Record Management System:

- a. No more than 65535 records can be used in a single data base file.
- b. No more than 32 fields can comprise any single record.
- c. No more than 254 characters can be placed in any one field.
- d. No more than 1000 characters can be placed into any single record.
- e. No more than 64 variables can be located in working storage at any given time.

2. Hardware

Following is a list of development hardware constraints that affected the design of this system and consequently impose constraints on the implementation of the Individual Training Record Management System.

- a. Main memory storage (RAM) is 512K.
- b. Two double density double sided disk drives were used for a total of 720K bytes of disk storage.

c. The IBM PC 8088 microprocessor operates at 4.7 MHz.

To maximize the amount of data that the system maintains, a RAM disk drive is created. This RAM drive allows the program to be copied into memory and enables the full use of the two disk drives for data. Use of the RAM drive increases the speed of the program by allowing the various called routines to be read more quickly into the working storage. The memory access time for the IBM PC is much faster than disk access time. This increase in speed reduces the response time to the user. The objective of the prototype is to handle the data required to maintain a battalion sized unit's training records. The amount of storage required for 800 individuals is approximately two megabytes. This allows for the overhead that the DEMS uses and the room necessary to execute functions that require the creation of temporary files. The available storage in the prototype system limited the test data base to a company sized unit.

3. Environmental

The following environmental considerations constrain the system by placing added requirements on the ultimate design.

- a. Inexperienced users
- b. Frequent turnover of personnel
- c. Limited training time available
- d. Hardware maintenance must be readily available from outside commercial sources.
- e. Software Maintenance is not intended to be performed at the user level because computer programming skills are not necessarily available.

f. No unusual requirements for physical environment such as smoke or dust free atmosphere, unusual electrical power characteristics or hookup, or vibration dampening, are required.

B. GENERAL

The outer layer consists of six separate modules divided into three levels. This section will briefly describe the routines that make up each module and the system output from each routine. The listings of the programs are in the appendices, and can be examined for further detail.

1. ITRMS Master Control Module

a. Master Routine (Master.prg)

This module controls the sequencing of the program. It is an endless loop which requires the computer to be either turned off or re-booted to exit the ITRM System. The Master routine requires the user to switch diskettes. The user receives a message to physically exchange the program diskettes with the data diskettes. If the diskette exchange is not made or is made improperly the user is prompted again to make the proper exchange.

The Access Module is initiated by the Master Routine and if a valid indication is returned, the Master Routine will initiate the Command Determiner Module. Once the Command Determiner Module is exited, the Master Routine loops to the beginning to receive a user command to initiate the Access Module.

b. Pause Routine (Pause.prg)

This routine provides a time delay for the system. When this routine is called, it delays the functioning of the system for approximately thirty seconds.

This routine is used to delay error messages that are displayed for a short period of time.

2. Access Module

a. Access Routine

The Access Routine prompts the user for an identification code. The routine then instructs the DBMS to search for that code in the security data base file. If the code is not found a message is displayed and the user can try again. If the code is found the corresponding password from the data base is stored in memory. The user is then prompted for the password. When it is entered, if it does not match the password in memory an error message is displayed and the user can try again. If the password that the user enters matches the password in memory, a valid user state is passed to the Master Routine with a user authorization level.

3. Command Determiner Module

a. Command Determiner Routine (Cmddet.prg)

The Command Determiner Routine displays to the user a menu. The user can choose to initiate the Nonstandard Report Generator, Standard Report Generator, Data Entry and System Functions (Maintenance), Help, or Quit. The user enters the appropriate letter which is processed by a case statement that initiates the selected module. Figure 4.1 displays the Command Determiner Menu.

b. Help2 Routine (Help2.prg)

The Help2 Routine provides information to the user on the functions that can be initiated from the Command Determiner and the Maintenance Module. The information is presented to the user to help in the selection of the

USMC INDIVIDUAL TRAINING
RECORD MANAGEMENT SYSTEM

MAIN MENU

SELECT ONE OF THE FOLLOWING OPTIONS BY
ENTERING THE APPROPRIATE LETTER

D...DATA ENTRY AND SYSTEM FUNCTIONS
S...STANDARD REPORTS
N...NONSTANDARD REPORTS
H...HELP
O...QUIT

SELECT OPTION =====>:

Figure 4.1 Command Determiner Menu

appropriate module and is not designed to explain the functioning of the entire system. The systems built-in prompts walk the user through the necessary steps to accomplish the desired task.

4. Nonstandard Report Generator Module

a. Nonstandard Report Routine (Nostdrpt.prg)

This module gives the user an option to go directly to the report or to view a help module. If the user desires to go directly to the report, this routine queries the user for the first set of attributes and attribute descriptions. After the user's selections are obtained, the routine searches the appropriate data base files to locate all records that satisfy the user specified attribute and attribute description. Attributes are field names such as RANK, COMPANY, RFLQUAL, etc. Up to three attributes from any data base files (the data base file location is invisible to the user) may be specified by the user. Multiple attributes are automatically joined by the routine using a logical "and" operation. The routine

employs a series of case statements. For each attribute selected, the attribute name is stored. Slctdesc.prg is called which stores an attribute description such as E5 (for RANK) or A CO (for COMPANY) or EX (for RFLQUAL). After the user has finished selecting attributes and attribute descriptions, the routine identifies the correct database files and either creates a temporary file containing the needed information or, if only two database files are involved, primary and secondary files are identified. The report is then displayed. Figure 4.2 depicts the sequence of screen displays for attributes and descriptions of "COMPANY is A CO" .AND. "RANK is E5" .AND. "PLATOON is 1ST PLAT".

```

NONSTANDARD REPORT GENERATOR
LCCATE ALL MARINES THAT SATISFY THE FOLLOWING;

<ATTRIBUTE>                                <DESCRIPTOR>
-----
COMPANY      IS      A CO
RANK         IS      E5
PLATOON      IS      1ST PLAT
PRESS 'ENTER' TO CONTINUE

```

Figure 4.2 Attributes and Descriptors Display

b. Nonstandard Help Routine (Nostdhlp.prg)

This routine responds to the user's request for "HELP" while in the Nonstandard Report Menu. It displays a brief explanation of the attribute and attribute description process.

c. Select Attribute Routine (Slctarti.prg)

This routine prompts the user to select attributes. After the first and second selection the user is asked if he wants to select another attribute (only three attributes are permitted, therefore after the third attribute and attribute description set have been selected the routine automatically executes the user request). If the response is yes, the selection procedure is repeated. This process is continued until the user answers no or three attributes have been selected. The routine locates the appropriate data base files and displays the NAME, SSN, and attribute information for each record that satisfies the user's specified criteria.

d. Select Descriptor Routine (Slctdesc.prg)

This routine prompts the user to select attribute descriptions. Each attribute description is stored into a memory variable for later use when the database files are searched. An attribute description is selected for each attribute.

5. Standard Report Generator Module

a. Standard Report Routine (Stdrrpt.prg)

This routine generates the primary menu for standard report selection and stores the user's response in a memory variable. Control is then passed to the appropriate subroutine to obtain the necessary data from the database. This routine will continue until the user responds with a "Q" (QUIT). Figure 4.3 displays the Standard Report Menu.

STANDARD REPORT SELECTION
SELECT REPORT BY ENTERING THE APPROPRIATE LETTER

A...UNIT ROSTER
B...TRAINING STATUS REPORT
C...TRAINING ROSTER REPORT
D...INDIVIDUAL TRAINING RECORD
E...MARKSMANSHIP STATUS REPORT
F...MARKSMANSHIP ROSTER REPORT
G...MOS STATUS REPORT
H...MOS ROSTER REPORT
I...PERSONAL DATA REPORT
J...EST STATUS REPORT
K...EST ROSTER REPORT
L...SWIM QUALIFICATION REPORT
M...HELP

Q...QUIT

SELECT OPTION =====>:

Figure 4.3 Standard Report Menu

b. Select Unit Routine (Slctunit.prg)

This routine prompts the user to select the size and name of the unit (i.e., battalion, company, platoon) for the report generation. The user responds with a letter that corresponds to a user installed company or platoon designation.

c. Unit Roster Routine (Unitrost.prg)

This is a control routine which creates the personnel roster for the unit specified by the user's response to the Slctunit Menu. Figure 4.4 displays a copy of the Personnel Roster Report.

d. Training Status Routine (Trngstat.prg)

For the selected unit, this routine calculates the total number of training elements completed for each training category. For example, the training category

UNIT ROSTER		
NAME	RANK	PRIMARY MOS
BAKER, STEVE N.	E4	0311
CAFY, R. V.	E5	0311
CASTIN, D. P.	E1	0311
DCNNELY, R. G.	E6	0331
PRESS 'ENTER' TO CONTINUE		

Figure 4.4 Personnel Roster Report

"HISTORY" consists of three training elements (A, B, C). A percent completed figure is calculated for each training element by dividing the total training elements completed (EX. ZZAHIS, ZZBHIS, ZZCHIS) by the total number of personnel in the unit (ZZTOTAL). Both the totals for each element and the percent completed figures are shown (see Figure 4.5).

e. Training Count Routine (Cntest.prg)

This routine counts the number of essential subject training elements that have been completed in the unit or subunit specified.

f. Qualification Count Routine (Cntqual.prg)

This routine saves all memory variables in a memory file (ZZTOTAL) and then counts the number of persons that have not completed SWIMQUAL, RFLQUAL, PSTQUAL and persons in the unit not required to qualify (NA).

TRAINING STATUS REPORT

CCDE CF CONDUCT...	42 PERCENT COMPLETED	A...	75
	80 PERCENT COMPLETED	B...	144
	48 PERCENT COMPLETED	C...	86
MARINE CORPS HISTORY...	73 PERCENT COMPLETED	A...	131
	82 PERCENT COMPLETED	B...	147
	66 PERCENT COMPLETED	C...	119
CLCSE ORDER DRILL...	46 PERCENT COMPLETED	A...	83
	64 PERCENT COMPLETED	B...	115
	79 PERCENT COMPLETED	C...	142

PRESS 'ENTER' TO CONTINUE

Figure 4.5 Training Status Report

g. Training Roster Routine (Trngrost.prg)

This routine provides the Training Roster Report. It starts at the top of the Pers.dbf and looks at each record individually. If a record corresponds to the unit selected by the user, the routine Trngrpt.prg is called which collects and displays the training data that corresponds to the current Pers.dbf record (e.g., If the user wants training data on every member of 1ST PLAT, A CO, Trngrost.prg locates each record in the Pers.dbf which belongs to 1ST PLAT, A CO. Trngrpt.prg then displays the training data from the appropriate data base files.). Information on one individual is displayed in six rows. Four sets of data are displayed per screen. The user presses the "enter" key to scroll forward to each succeeding screen display.

h. Training Roster Format Routine (Trngrpt.prg)

This defines a coded report format which displays COC, HIS, MKS, COD, INT, TAC, NBC, PFT1CLSS, PFT2CLSS, UNI and AID for each member of the unit or subunit selected.

i. Marksmanship Status Routine (Mksstat.prg)

This routine creates a temporary file which contains the current year rifle and pistol qualification results for each member in the selected unit or subunit. The results are summed by qualification category for the rifle and pistol. Percentages are calculated and displayed in a summary status report which is automatically displayed on the screen. From this module a simple decision support submodule may be called at the discretion of the user. If the user desires to analyze his remaining rifle range quotas and compare them with his total number of range requirements for the current year, he may elect to "Analyze Rifle Range Quotas" (See Quota Study Routine for a more complete description.). Figure 4.6 displays the Marksmanship Status Report.

j. Convert Unit Routine (Cvrtunit.prg)

This routine stores the name of the company and platoon selected (unit and subunit) by the user during routine 'Slctunit'. This is necessary because when the user selects a unit and subunit, it is done by menu selection. Accordingly the only thing stored into memory is the corresponding letter from the menu provided by Slctunit.prg (A, B, C, D, E). Some screen displays require the unit name (e.g., "A CC" vice "A") so a conversion is necessary.

MARKSMANSHIP STATUS REPORT

RIFLE EXPERT...	48	PISTOL EXPERT...	3
RIFLE SHARPSHOOTER...	36	PISTOL SHARPSHOOTER...	0
RIFLE MARKSMEN...	48	PISTOL MARKSMEN...	2
UNQUALIFIED...	16	UNQUALIFIED...	0

PERCENTAGES

RIFLE EXPERT...	32	PISTOL EXPERT...	60
RIFLE SHARPSHOOTER...	24	PISTOL SHARPSHOOTER...	0
RIFLE MARKSMEN...	32	PISTOL MARKSMEN...	40
UNQUALIFIED...	11	UNQUALIFIED...	0

PRESS 'ENTER' TO CCNTINUE

Figure 4.6 Marksmanship Status Report

k. Quota Study Routine (Qtastudy.prg)

The Qtastudy.prg and the Quotapad.prg make up the decision support submodule. This module is used by Mkssstat.prg and displays the range quotas that remain for the current year. The current month is extracted from the log on date. The range quotas for each unit are stored in a memory file during the annual initialization procedure. The appropriate set of range quotas for the unit involved is retrieved and stored into memory variables. Summary information is provided concerning the relationship between remaining range quota requirements and remaining range quota allocations. Upon conclusions of the module the user may return to the Standard Report Menu or to a scratch pad to revise quota assignments. Revised quota assignments from the scratch pad can either be permanently saved in the range quota memory file or ignored. See Appendix B for a more detailed description.

1. Quota Scratch Pad Routine (Quotapad.prg)

This module gives the user an opportunity to interactively modify the remaining monthly rifle range quotas for his unit. This can be done an unlimited number of times. Each time the screen display returns to the actual monthly allocation as base data. After each iteration, the new set of range quotas is totaled and compared to the number of rifle range quotas required by the unit for the remainder of the year. A comment is displayed which indicates the net result (scratch pad quotas less than requirements, scratch pad quotas greater than requirements, or scratch pad quotas equal to requirements). However, if the user elects to permanently replace the monthly quota allocation in memory, he may do so by selecting the "Replace Current Quotas" option.

m. Marksmanship Roster Routine (Mksrost.prg)

This routine creates a temporary file which includes NAME, RFLQUAL, RFLDATE, PSTQUAL, PSTDATE for each member in the selected unit or subunit. A standard report format (B:Mrksrpt) is used to display the Marksmanship Roster Report (see Figure 4.7).

n. MOS Status Routine (Mosstat.prg)

This routine produces the MOS Status Report. Each type of MOS in the specified unit or subunit is counted and totaled, regardless of the type or number of different MOS's in the particular unit or subunit. The results are then displayed by MOS and by rank (see Figure 4.8).

c. MOS Roster Routine (Mosrost.prg)

This is a control routine which creates the Personnel Roster for the unit or subunit specified by the user's response to the Slctunit Menu (see Figure 4.9).

MARKSMANSHIP ROSTER REPORT

NAME	RANK	QUAL DATE	QUAL
EAKER, STEVE N.	E4	100184	EX
CAFY, R.V.	E5	050184	MM
CASIN, D.P.	E1	050184	UN
DCNNELY, R.W.	E6	DDMMYY	**

PRESS 'ENTER' TO CONTINUE

Figure 4.7 Marksmanship Roster Report

MOS STATUS REPORT A CO, 3RD PLAT

MCS	RANK	TOTAL
0311	E1	8
0311	E2	12
0311	E3	9
0311	E4	8
0311	E5	3

Figure 4.8 MOS Status Report

f. Personnel Data Routine (Persdata.prg)

This is a control routine which creates the Personnel Roster for the unit or subunit specified by the user's response to the Slctunit Menu.

g. Personnel Data Format Routine (Mosrpt.frm)

This defines a dBASE II report format which has already been determined and is stored in memory.

MOS ROSTER REPORT

NAME	RANK	PRIMARY MOS	SEC MOS
* MOS 0311			
BAKER, STEVE N.	E4	0311	0331
PCILAFD, R.W.	E2	0311	****
CARY, R.V.	E5	0311	****
PHILLIP, D.P.	E4	0311	****

PRESS 'ENTER' TO CCNTINUE

Figure 4.9 MOS Roster Report

r. EST Status Routine (Eststat.prg)

This routine creates a temporary file that includes all essential subject and PFT fields from the Est.dbf for members in a user specified unit or subunit. From this temporary file the routine counts the total number of unit members that have completed all elements in each training category. The EST Status Report is displayed and includes totals for each training category and the percent completed (see Figure 4.10).

s. EST Count Routine (Cntest.prg)

This is a short routine that counts the total training categories containing an "*" which indicates a training element has not been completed (i.e., It counts the number of training categories that currently include one or more training elements that have not been completed).

t. EST Roster Routine (Estrost.prg)

This is a control routine which creates the Personnel Roster Report for the unit or subunit specified by the user's response to the Slctunit Menu.

EST STATUS REPORT

CODE OF CONDUCT...	42 PERCENT COMPLETED...	76
MARINE CORPS HISTORY...	84 PERCENT COMPLETED...	151
CLOSE ORDER DRILL...	39 PERCENT COMPLETED...	70
INTERIOR GUARD...	63 PERCENT COMPLETED...	113
FIRST AID...	8 PERCENT COMPLETED...	14

PRESS 'ENTER' TO CONTINUE

Figure 4.10 EST Status Report

u. EST Report Format Routine (Estrpt.prg)

This routine formats and displays the EST Roster Report. It is called by Estrcst.prg.

v. Swim Qualification Routine (swimqual.prg)

This routine displays the Swim Qual Report for the selected unit or subunit. The Swim Qual Report consists of each Marine's NAME, RANK, UNIT, and swimming qualification status. At the end of the report, summary data is displayed which includes the totals for each qualification category and the overall total for the unit or subunit. Figure 4.11 is a copy of the Swimming Qualification Status Report.

w. Swim Report Routine (Swimrpt.prg)

This routine, called by Swimqual.prg counts and stores the number of personnel in the selected unit or subunit that are in each swimming qualification category. Also, this routine displays the NAME, RANK, COMPANY and swimming qualification for each member of the unit or subunit.

BAKER, STEVE N. SWIM QUAL S3	E4	A CO
CARY, R.V. SWIM QUAL S2	E5	A CO
PHILLIP, D.P. SWIM QUAL WS	E4	A CO

SUMMARY DATA

SWIM QUALIFICATION	TOTAL
WS	3
S1	36
S2	54
S3	49
UNQUALIFIED	36

---T78---

PRESS 'ENTER' TO CONTINUE

Figure 4.11 Swimming Qualification Status Report

x. Help Routine (Help.prg)

This routine provides a functional description of each report listed in the Standard Report Selection Menu.

6. Maintenance Module

a. Maintain Control Routine (Maintain.prg)

The Maintain Control Routine produces a menu, Figure 4.12, that allows the user to select the desired function. This module restricts access when the function has a critical capability. Only the Data Base Administrator (DBA) and the Assistant DBA have access to all functions.

b. View Control Routine (Viewitr.prg)

This module queries the user for the name of the individual whose training record is to be viewed. The


```
MAINTENANCE MODULE MENU  
SELECT ONE OF THE FOLLOWING OPTIONS  
V...VIEW INDIVIDUAL DATA  
U...UPDATE DATA  
C...CREATE AN ITR  
D...DELETE AN ITR  
S...SYSTEM FUNCTIONS  
H...HELP  
*Q...QUIT*  
SELECT OPTION ==>:
```

Figure 4.12 Maintenance Module Menu

routine controls the search and creation of the formatted screen that displays the ITR. The routine asks the user a series of questions that can be answered with yes or no. This routine allows only the data to be displayed (see Figure 4.13).


```

INDIVIDUAL TRAINING RECORD

NAME:LAST, FIRST MI.      : GRADE:      MOS:      I/:      I
BIRTHDATE.....:DDMMYY:  SSN.....:
CO:      :PLAT:      :JOIN:DDMMYY: EAS.:DDMMYY:GAS MASK...I I ,
MIL CODE OF CONDUCT:****:  FF11.....: 0:/:  DATE:DDMMYY:
HISTORY OF MARCOR...:****:  FF12.....: 0:/:  DATE:DDMMYY:
DRILL.....:****:  NRC.....:****:
INTERIOR GUARD.....:****:  MARKSMANSHIP.....:****:
FIRST AID.....:*****:INDIVIDUAL TACTICAL
EQUIP & UNIFORM.....:****:  MEASURES.....:*****:

SWIM QUAL...S : DATE:DDMMYY:  WEIGHT CONTROL.....:
RIFLE QUAL...:000:/:  DATE:DDMMYY:  WEIGHT...:000:
PISTOL QUAL...:000:/:  DATE:DDMMYY:  HEIGHT...:00:

ELECTRONIC WARFARE...:*****:  DRUG ABUSE.....:*****:
OLD WEATHER.....:*****:  ALCOHOL ABUSE.....:*****:
LAW OF WAR.....:*****:  HUMAN RELATIONS...:*****:
TRAINING NOS.....:*****:  PERSONAL AFFAIRS...:*****:
LEADERSHIP.....:*****:  UCMJ.....:*****:
CHAR & MORAL ED.....:*****:
COMMENTS.:F:

```

Figure 4.13 ITR Screen Format

c. ITR Screen Format Routine (Itrscrn.prg)

This routine produces the electronic ITR on which an individual's data is displayed.

d. Screen Data Retrieval Routine (Getdata.prg)

The Getdata.prg communicates with the DBMS and retrieves the data for a particular individual. It then sends the data to the appropriate position on the screen that corresponds to the ITR format. The DBMS locates individuals by their SSN. Four data base files are used to create an ITR; Pers.dbf, Est.dbf, Qual.dbf, and Infomisc.dbf.

e. Comment Retrieval Routine (Getcmmnt.prg)

If unstructured comments exist for an individual, the routine will instruct the DBMS to search for them. The comment information will then be displayed on the screen.

f. Update Control Routine (Updtitr.prg)

The Update Control Routine questions the user concerning which ITR is to be updated. The routine determines if the individual is in the data base and controls the update of the ITR. The routine allows the user to update as many ITR's as desired. The data is displayed on the screen in the ITR format and allows a modified form of full screen editing.

g. Data Update Routine (Updata.prg)

This routine requires the user to type in the name of the field to be updated. The routine then highlights that field and permits data to be entered. When the data has been entered this routine initiates error checking

and initiates calculations. The data is then stored in the data base.

h. ITRM Creation Routine (Creatitr.prg)

The creation of a new ITR for an individual is controlled by this routine. It initiates the formatting of the screen and controls the routine which gathers the data for the individual. The routine allows multiple records to be created.

i. ITR Data Initialization Routine (Indata.prg)

This routine initializes the data base and allows the user to enter data for an individual by field. Once the data has been entered the module initializes error checking and calculations that may pertain to the data fields. If unstructured comments are to be entered, the routine controls this process. The ITR Data Initialization Routine instructs the DBMS to store the data in the appropriate data base files.

j. Data Error Checking Routine (Error.prg)

This routine performs the error checking for the system during data entry. Since this is a prototype system, error checking is not extensive. Dates, value ranges, and acceptable characters are the limits of the error checking. Dates are only checked for limits in number of months, days in the month, and acceptable year window. Specific months are not matched to specific number of days. The system does catch the most common errors caused by gross typing mistakes but may not catch slight errors if the values are within specified limits.

k. Calculation Routine (Calculat.prg)

The Calculation Routine performs the calculations necessary to reproduce training tables concerning weight/height standards, physical fitness standards by age, rifle qualification levels , and pistol qualification levels. When the information that is needed for these calculations has been entered in the data base the system recalculates the related fields and enters the new qualifications in the data base.

l. Comment Update Routine (Wrtcmmt.prg)

This routine finds the individual's comment record in the data base and updates the information with new data. It then stores the data back into the data base.

m. Comment Creation Routine (Crtcmmt.prg)

The Comment Creation Routine creates an unformatted comment record for an individual in the data base. The comment record is filled with asterisks that will be typed over when data is inserted.

n. Delete ITR Routine (Deletitr.prg)

This routine is limited to the DBA and Assistant DBA only. The routine queries the user for the name of the individual to be deleted. It then finds the individuals ITRM information and displays the data in ITR format. A second verification, that this individual is to be deleted, is obtained at this time. The module then deletes the individual from all the data base files. This deletion is permanent and there is no recovery of deleted records.

c. System Function Control Routine (System.prg)

The control of system functions is accomplished by this routine. Like all previous routines, this routine is menu driven (see Figure 4.14). The user selects the menu option that performs the desired function, and calls the appropriate routine.

```
SYSTEM FUNCTION MENU
SELECT ONE OF THE FOLLOWING OPTIONS
I...INSTALL SYSTEM
S...SYSTEM RESET (YEARLY)
A...ACCESS LIST MAINTENANCE
Q...QUIT TC MAINTENANCE MENU
```

```
ENTER OPTION ==>:
```

Figure 4.14 System Function Menu

p. System Installation Routine (Install.prg)

Units in the Marine Corps often have different names. The system allows the DBA to name his units to correspond to the actual names of the battalion's units. The system also initializes the current year. This information is then stored on a file which is recalled whenever the system is functioning.

g. System Reset Routine (Sysreset.prg)

Training in the Marine Corps is based on a yearly cycle and requires that individuals requalify in certain areas once or more a year. This routine resets globally all the fields in the data base that must be

requalified each year. The routine allows the DBA to enter the yearly range quotas. The DBA can use this routine to update those quotas, but this is not recommended. The capability also exists in the Standard Report Generators DSS.

r. System Access Control Routine (Rstaccss.prg)

The DBA must be able to change the access list by deletions or additions. This routine displays a menu, (Figure 4.15) that gives the DBA options to choose. This routine gives the DBA the ability to manage access to the system and maintain a list of users. It is not recommended that the DBA keep a printed copy of this list unless it is in a secured environment. The system will list each user, user identification code, user password, and authorization level.

```
          SECURITY ACCESS LIST
CHOOSE OPTION TO BE EXECUTED

I...LIST ALL USERS
A...ADD TO ACCESS LIST
D...DELETE FROM ACCESS LIST
Q...QUIT TO SYSTEM FUNCTION MENU

ENTER OPTION ====>:
```

Figure 4.15 Access Function Menu

V. CONCLUSION

A. GENERAL

The thesis concept statement is written in general terms and therefore, is subject to qualitative evaluation only. This approach is intended for the following reason. During the conception phase of the project, we had only a vague intuitive "feel" for such things as reasonable system costs, acceptable response times, operator training requirements, and the ultimate "user friendliness" that could be achieved. Because functional requirements are not well defined at this point, an attempt to impose stringent quantitative standards is premature. Furthermore, quantitative goals shift the emphasis toward achieving system efficiency and away from defining system effectiveness. In other words, the purpose of this thesis is to prove a general concept through the development of a working prototype. The general concept includes simplicity of operation and maintenance. System efficiency is not a significant consideration. The reader will notice, however, that system specifications are defined using quantifiable objectives where feasible, the distinction being that the system requirements determined by the user may not be satisfied by the first generation prototype. As the user becomes more familiar with the capabilities of the system, requirements will become more clearly defined. With this approach the overall system concept may be valid while some system requirements have not yet been attained (i.e., in the first generation prototype).

E. SYSTEM SPECIFICATIONS (FUNCTIONAL)

All system specifications, except those referred to below, have been satisfied by the prototype.

1. Storage Limitations

The prototype cannot adequately handle the specified 800 records. The limiting constraint is available storage. All system software resides on four double-sided, double-density 5 1/4 inch floppy disks (360k bytes each). The first two disks are program disks which are stored directly into RAM memory. The second two disks contain all of the data base files. Data base files are read into working storage only when they are being used by the system. Each Marine in the data base requires approximately 550 bytes of data (or 4.4k bits). Due to the DBMS's required overhead the prototype can accommodate approximately 180 Marines. In order to achieve the specification requirement of 800 Marines, a storage device capacity of two megabytes or more is required (e.g., hard disk drive).

2. Retrieval Limitations

All standard reports can be retrieved by unit or subunit. However, the standard report does not include the facility to retrieve data by other user defined attributes. This facility is, however, included in the nonstandard report. The nonstandard report may include up to three user defined attributes and attribute descriptions.

C. SYSTEM SPECIFICATIONS (NON-FUNCTIONAL)

1. Training

Most of the non-functional specifications relate to the operator training effort required to implement the

system. It is our judgement that the prototype satisfies these requirements.

2. Error Checking

Operator input is range checked, data type checked, and checked for field length (characters in excess of the maximum field length are truncated). In a fully implemented system error checking would be expanded and diagnostic error messages sent to the user.

3. Table Mapping

Mapping of rifle, pistol and PFT scores into qualification categories is invisible to the user. In addition, a weight control table is embedded. When an individual's height and weigh are entered the weight control table is referenced. If his weight is in excess of allowable standards for his height, he is automatically assigned to the "weight control program" by assigning a boolean variable of "T" to the weight control field in his training record.

D. SUBSEQUENT SYSTEM DEVELOPMENT

The next step in the development of the system is to place the system into the hands of the user to accomplish the following objectives:

- a. Define user training requirements;
- b. Achieve user familiarity;
- c. Evaluate and quantify existing functional requirements;
- d. Define additional quantifiable functional requirements;

e. Design and develop second generation prototype based on new set of user defined functional requirements.

E. SUMMARY

The objective of the Thesis was to prove the feasibility of a concept. It is reasonable to conclude that the concept has been proven through the design, the implementation, and the demonstration of a working prototype.

APPENDIX A

DATA ELEMENT DICTIONARY

All data elements that exist in the individual training data management system are contained in one of seven data base files. The only exception is the data element SSN. SSN is a key field and is therefore included in the record structure of multiple files. The following information is used to describe each data base file:

A. FIELD

Each data element belongs to a specific field within the record structure. Fields are numbered sequentially. For example, field 03 in the data base file called Pers is the data element named Rank.

E. NAME

Each data field has a unique name which always refers to that specific information that is contained in the data field (example; PrimeMOS contains the primary MOS of each member in the data base).

C. TYPE

Each data field is defined to be one of three possible data types:

1. Character (C)
2. Numeric (N)
3. Logical (L)

D. WIDTH

This characteristic describes the maximum length of the data field. Information in the data field may not exceed the number of spaces that have been set aside for that particular field (all characters and blanks are included when counting the number of spaces in a field).

E. DESCRIPTION

This column contains a narrative description of the field. In many instances, examples (Ex.) are used to describe the structure of a typical data element.

F. DATA BASE FILES

1. Pers (Personnel)

Field	Name	Type	Width	Description
01	SSN	C	09	Ex. 240707765
02	Name	C	30	Last name followed by a comma, First name and Middle Initial. (Ex. Smith, Jack E.)
03	Rank	C	02	Pay grade (Ex. E2, O1)
04	Primemos	C	04	Primary Military Occupational Specialty (Ex. 0311)
05	Secmos	C	04	Secondary MOS
06	Ccompany	C	06	Ex. A CO, HHS CO, A BTRY
07	Platoon	C	09	Ex. 1ST PLAT, GUN PLAT, WPSN PLAT, HDQTRS
08	Joindate	C	06	The date (DDMMYY) member joined the unit.
09	EAS	C	06	Expiration of Active Service (DDMMYY)
10	Erthdate	C	06	Date of birth (DDMMYY)
11	Ccomment	L	01	A boolean variable (F/T) for which T indicates that a comment record exists.
12	Height	C	02	Individual's height in inches
13	Weight	C	03	Individual's weight in pounds
14	Wtcnt	L	01	A boolean variable (F/T) for which T indicates that the individual is on weight control.
15	Gasmask	C	01	Gas mask size (S, M, L)

2. FST (Essential Subjects Testing)

Field	Name	Type	Width	Description
01	SSN	C	09	EX. 240707765
02	COC	C	03	Code of Conduct contains an entry for each training element (A,B,C).
03	HIS	C	03	History of the Marine Corps contains an entry for each training element (A,B,C).
04	COD	C	03	Close Order Drill contains an entry for each training element (A,B,C).
05	INT	C	03	Interior Guard contains an entry for each training element (A,B,C).
06	AID	C	11	First Aid contains an entry for each training element (A,B,C,D,E,F,G,H,I,J,K).
07	UNI	C	03	Equipment & Uniforms contains an entry for each training element (A,B,C).
08	EFT1RAW	N	03	Includes the numeric score for the first annual physical fitness test (RANGE:00-300).
09	EFT1DATE	C	06	Date of first physical fitness test (DDMMYY).
10	EFT1CLSS	C	01	Results of first physical fitness test (1,2,3,U).
11	EFT2RAW	N	03	Includes the numeric score for the second annual physical fitness test (RANGE:00-300).
12	EFT2DATE	C	06	Date of the second physical fitness test.
13	EFT2CLSS	C	01	Results of second physical fitness test (1,2,3,U).
14	NBC	C	06	NBC contains an entry for each training element (A,B,C,D,E,F).
15	MKS	C	06	Marksmanship contains an entry for each training element (A,B,C,D,E,F).
16	TAC	C	05	Individual Tactical Measures contains an entry for each training element (A,B,C,D,E).

3. QUAL (Qualifications)

Field	Name	Type	Width	Description
01	SSN	C	09	EX. 240707765
02	SWIMQUAL	C	02	Swimming Qualification (WQ, S1,S2,S3).
03	SWIMDATE	C	06	SWIMQUAL date (DDMMYY).
04	RFLSCORE	C	03	Numeric rifle score (RANGE:000-250).
05	RFLQUAL	C	02	Rifle Qualification (EX,SS,MM,UN).
06	RFLDATE	C	06	Date of RFLQUAL (DDMMYY).
07	PSTSCORE	C	03	Numeric pistol score (Range:000-300).
08	PSTQUAL	C	02	Pistol Qualification (EX,SS,MM,UN).
09	PSTDATE	C	06	Date of PSTQUAL (DDMMYY).

4. INFCMISC (Information Miscellaneous)

Field	Name	Type	Width	Description
01	SSN	C	09	EX. 240707765
02	EIECWAR	C	06	Electronic Warfare contains an entry for each training element (A,B,C,D,E,F).
03	CLDWTHR	C	06	Cold Weather Training contains an entry for each training element (A,B,C,D,E,F).
04	IWWAR	C	06	Law of War contains an entry for each training element (A,B,C,D,E,F).
05	MOS	C	04	MOS Training contains an entry for each training element (A,B,C,D).
06	IDRSHP	C	06	Leadership Training contains an entry for each training element (A,B,C,D,E,F).
07	DRUG	C	06	Drug Abuse contains an entry for each training element (A,B,C,D,E,F).
08	ALCOHOL	C	06	Alcohol Abuse contains an entry for each training element (A,B,C,D,E,F).
09	HUMREL	C	06	Human Relations contains an entry for each training element (A,B,C,D,E,F).
10	PERSAFFR	C	06	Personal Affairs contains an entry for each training element (A,B,C,D,E,F).
11	UCMJ	C	06	UCMJ contains an entry for each training element (A,B,C,D,E,F).
12	CHRMORED	C	06	Character and Moral Education contains an entry for each training elements (A,B,C,D,E,F).

5. SECURITY (Access Information)

Field	Name	Type	Width	Description
01	USERID	C	05	Numeric code that uniquely identifies each user.
02	USERPASS	C	08	Alphanumeric code that uniquely identifies each user.
03	AUTHLEV	N	01	Numeric code that defines authorization level for each user.
04	NAME	C	15	Alphanumeric that identifies the user.

6. CCMMMENTS (Non-formatted Information)

Field	Name	Type	Width	Description
01	SSN	C	09	EX. 240707765
02	INFCTXT1	C	25	Unstructured comments.
03	INFCTXT2	C	25	Unstructured comments.
04	INFCTXT3	C	25	Unstructured comments.
05	INFCTXT4	C	25	Unstructured comments.
06	INFCTXT5	C	25	Unstructured comments.
07	INFCTXT6	C	25	Unstructured comments.

08	INFCTXT7	C	25	Unstructured comments.
09	INFOTXT8	C	25	Unstructured comments.
10	INFCTXT9	C	25	Unstructured comments.
11	INFOTXT0	C	25	Unstructured comments.

APPENDIX B

DECISION SUPPORT SYSTEM (DSS) DESCRIPTION

A. BACKGROUND

The management of training records includes many routine tasks that are repetitious and highly structured. Many of these tasks incorporate low level, well-defined or semi-structured, independent decision making processes. Heretofore, these tasks have been manually implemented at the cost of many manhours, and sometimes decisions have been made without making the best use of available information. Too often the result has been wasted effort, frustration, and incorrect decisions.

B. PURPOSE

The purpose of this submodule is to demonstrate the application of a "first generation" prototype DSS that will achieve the following:

- a. Demonstrate typical capabilities of DSS in the training management environment.
- b. Demonstrate the small knowledge base necessary to implement a properly designed DSS.
- c. Serve as a "functioning prototype" to gain early user support and experience in order to further refine system requirements.

C. FUNCTIONAL DESCRIPTION

The DSS element of the software system product is implemented in the "Marksmanship Status Report" routine which is

called by the "Standard Report Generator". When the user selects the "Marksmanship Status Report" from the "Standard Report Menu" the DSS submodule is implemented. The DSS submodule performs the following functions at the company level:

a. Calculates and displays a current "Marksmanship Status Report" which includes:

(1) A summary of rifle and pistol current year requalification data (number of experts, sharpshooters, marksmen, and unqualified).

(2) The total number of persons that have fired for qualification during the current year.

(3) The percent of experts, sharpshooters, marksmen and unqualified shooters based on the total number of people that have fired for qualification during the current year.

b. A company rifle range quota analysis may then be performed at the user's discretion. This routine includes the following functions:

(1) Calculates and displays a range quota analysis.

(2) The following summary information is displayed:

(a) Total number of Marines fired for qualification during current year.

(b) Total number of Marines fired for qualification during current year, but failed to qualify.

(c) Total number of Marines not required to fire for qualification during the current year.

(d) Total number of Marines to qualify before the end of the current year.

(3) A chart is displayed which shows the months remaining in the current year, the number of range

quotas allocated for each month, and the total number of rifle range quotas that remain for the current year.

(4) As appropriate, one of the following elements of information is displayed:

(a) Additional quota requirements for the year

(b) Excess quotas allocated for the year

(c) Quota requirements (number of person left to qualify) equal quota allocations.

c. User's may manipulate allocations on a scratch pad to visualize different combinations of monthly quota allocations or to determine their effect on end of year results (the objective is to manage quotas so that quota requirements equal quotas allocated). If desired, a set of "scratch pad" range quotas can permanently replace the unit's range quota allocations for the remainder of the current year. If this option is elected all subsequent calculations will be based on the "new set of range quota allocations".

D. USER'S VIEW

1. Summary Information

Decision making is supported by providing summary information for the company. This snapshot of a unit's current marksmanship status demonstrates the degree of success of the unit's marksmanship training program. By comparing this summary to unit marksmanship objectives the system can be used as a control mechanism. The following types of decisions can be supported:

(1) Evaluate marksmanship training program for weaknesses as indicated by unsatisfactory qualification results to date.

(2) Evaluate unit qualification objectives for feasibility.

2. Support Depth

Decision making is supported at many levels from the platoon or company training NCO to the battalion operations officer.

3. Interdependence

Interdependent (sequential and pooled) decisions are supported. For example, if one company permanently changes rifle range quota allocations this decision is captured by the system and reflected in any subsequent analysis

4. Restrictions

Due to the overriding system requirement for limited user training all processes are menu driven. This restricts the degree of user control and flexibility. This is a significant compromise which reduces its effectiveness as a DSS.

E. BUILDER'S VIEW

1. Dialogue

The dialogue subsystem (implemented in dBase II) creates the user's view as discussed above. It performs all necessary data base functions, extracts and processes summary information.

2. Modelling

The model base subsystem includes the mathematical and process models necessary to use the "scratch pad" utilities. For example, a model file is constructed in memory for each unit which has allocated quotas. Quotas can be

selectively extracted, compared, permanently changed by the user (provided the user has the appropriate system access authorization). Models are not integrated, but each model is fully integrated with the DBMS through the query language. Integration of models is a desirable feature (e.g., battalion should have the capability to total the quotas of all companies on a monthly basis or on a remaining current year basis) and could be implemented. This is the type of added requirement that should be defined by the user during operation of this "first generation" prototype.

F. CCNTENTS OF KNOWLEDGE BASE

Data Base files (DBF) contain qualification status of each member for the current year and members not required to fire for qualification during the current year.

Memory files (MEM) containing rifle range quotas for each company are stored in separate memory files (see Figure B.1 for example).

CORNGQTA.MEM		
VARIABLE NAME		VALUE
QJAN	-	25
QFEE		25
QMAR		25
QAPR		25
QMAY		50
QJUN		50
QJUL		50
QAUG		50
QSEP		00
QOCT		00
QNOV		00
QDEC		00

Figure B.1 Memory File

APPENDIX C

ITRS MASTER CONTROL MODULE LISTING

A. MASTER ROUTINE

```
* Routine Name: Master.prg
* Module Name: ITRS Master Control Module
* Version: 1.0
* Author: D.P. Haeusler
* Date: 28 Oct 83
* Variables Used: Thru, Valpass
* Variables Modified: None
* Variables Created: None
* Variables Released: None
* Files (opened/closed): None
* Temp Files Created: None
* Using Subroutines: None
* Description: This is the routine that controls the flow
* of the program. This routine is an endless loop which
* requires the user to turn the computer off in order to
* enter another system. This routine does not allow the
* users to directly enter the DBMS, the general intent is
* to make the DBMS completely transparent to users. Some
* house keeping is done to insure that the proper disks
* are in place. This occurs by reading stored files on
* the disks and then checking the values. The system will
* not continue until the disks are in the proper drive.
```

```
SET CCNSCLE ON
SET TALK OFF
SET ESCAPE CFF
```

```
STORE T TO AAENDLES
STORE F TO SWITCH
DO WHILE .NOT. SWITCH
```

```
    ERASE
    @ 9,25 SAY " REMOVE DISKETTES FROM THE DRIVES"
    @ 10,25 SAY " PLACE 'A' DATA DISK IN DRIVE 'A'"
    @ 11,25 SAY " PLACE 'B' DATA DISK IN DRIVE 'B'"
    @ 23,25 SAY "***** PRESS ANY KEY TO CONTINUE *****"
    SET CCNSCLE OFF
    WAIT
    SET CCNSCLE ON
    RESTORE FROM A:MEMDISK ADDITIVE
ENDDO
DO WHILE AAENDLES
    ERASE
    @ 10,27 SAY " UNITED STATES MARINE CORPS"
    @ 11,23 SAY " INDIVIDUAL TRAINING RECORD SYSTEM"
    @ 12,34 SAY " (PROTOTYPE) "
DO PAUSE
    @ 23,23 SAY "***** PRESS ANY KEY TO CONTINUE *****"
SET CCNSCLE OFF
WAIT
SET CCNSCLE ON
DO Access
RESTORE FROM A:MEMDISK ADDITIVE
STORE F TO SYSDATE
DO WHILE .NOT. SYSDATE .AND. AAVALPAS
    ERASE
```



```

@ 11,24 SAY "ENTER TODAY'S DATE"
ACCEPT "          DATE (DDMMYY) ==> " TO SYSDAY
IF VAL($ (SYSDAY,1,2)) < 32 .AND. VAL($ (SYSDAY,3,2)) ;
< 13 .AND. VAL($ (SYSDAY,5,2)) = CRNTYR
STCRE SYSDAY TC AADATE
STCRE T TO SYSLATE
ELSE
FRASE
@ 8,24 SAY "IMFROPER DATE--REENTER"
@ 9,24 SAY "***PRESS ANY KEY TO CONTINUE**"
SET CCNSOLE OFF
WAIT
SET CCNSOLE ON
ENDIF
ENDDO
RELEASE ALL EXCEPT AA??????
IF AAvalias
DO Cmoddet
ENDIF
STORE T TO AAENDLESS
ENDDC

```


E. PAUSE ROUTINE

```
* Routine Name: Pause.prg
* Module Name: ITRS Master Control Module
* Version: 1.0.0.0.1.0
* Author: D.F. Haeusler
* Date: 30 Nov 83
* Variables Used: pause
* Variables Modified: pause
* Variables Created: pause
* Variables Released: pause
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Access.prg, Maintain.prg, Master.prg
* Description: This routine creates a delay while a message
* is being displayed.
```

```
STORE 1 TO pause
DO WHILE pause < 75
    STORE pause + 1 TO pause
ENDDC
```

```
RELEASE pause
RETURN
```


APPENDIX D ACCESS MODULE LISTING

A. ACCESS ROUTINE

```
* Routine Name: Access.prg
* Module Name: Access Module
* Version: 2.0
* Author: D.P. Haeusler
* Date: 28 Oct 83
* Variables Used: aavalpass, password, idnum, validid,
*                 counter, thru, pause, passrec, aalevel
* Variables Modified: thru, aavalpass, validid, idnum,
*                 password, counter, pause, passrec,
*                 aalevel
* Variables Created: aavalpass, password, idnum, validid,
*                 counter, pause, thru, passrec,
*                 aalevel
* Variables Released: counter, pause, passrec, validid,
*                 thru
* Files (opened/closed): a:security (opened/closed)
* Temporary Files Created: none
* Using Subroutines: Master
* Description: This routine conducts the logon procedures
* to gain access into the system. The routine will query
* the user for a user ID and a Password, after three
* incorrect responses the session will be terminated.
*
* BEGIN Access

STORE f TO thru
STORE f TO Aavalpas
STORE " TO password
STORE " TO idnum
STORE f TO validid
STORE 0 TO counter

DO WHILE .NOT. validid .AND. .NOT. thru

  ERASE
  @ 12,25 SAY " ENTER ITRS USER NUMBER OR "
  @ 13,25 SAY " 'QUIT' TO TERMINATE SESSION "

  ACCEPT "          ENTER USER NUMBER ==> " TO idnum
  STCRE !(idnum) TO idnum

  IF idnum = 'QUIT'
    STCRE t TO thru
  ELSE
    ERASE
    USE A:SECURITY INDEX A:SECINDX
    FIND &IDNUM
    IF # = 0
      USE
    ELSE
      STORE TRIM (USERPASS) TO PASSREC
      STCRE AUTHLEV TO AALEVEL
      STCRE T TO VALIDID
    ENDIF
  ENENDIF
```



```

IF .NCT. validid .AND. .NOT. thru .AND. COUNTER <> 2
  STCRE counter+1 TO counter
  ERASE
  @ 15,25 SAY "INVALID USEF NUMBER"
  @ 17,25 SAY " CHECK YOUR NUMBER "
  DO pause
ELSE
  IF .NCT. VALIDID .AND. .NCT. THRU .AND. COUNTER = 2
    STORE CCOUNTER+1 TO COUNTER
  ENDIF
ENDIF
ENDIF

IF counter = 3
  STCRE t TO thru
  ERASE
  @ 12,20 SAY "*****"
  @ 13,20 SAY "***** TERMINATE ACCESS *****"
  @ 14,20 SAY "*****"
ENDIF

ENDDC

STORE 0 TO counter
DO WHILE .NCT. AAválpas .AND. .NOT. thru

  ERASE
  @ 7,25 SAY " ENTER YOUR PASSWORD OR "
  @ 8,25 SAY " 'QUIT' TO TERMINATE THE SESSION"
  @ 11,18 SAY " THE PASSWORD WILL NOT BE DISPLAYED "
  @ 14,25 SAY " ENTER PASSWORD ==> "

  SET CCNSCLE OFF
  ACCEPT " ENTER PASSWORD ==> " TO password
  SET CCNSCLE ON

  IF !(password) = 'QUIT'
    STCRE t TO thru
  ELSE
    IF !(password) = passrec
      STCRE t TO AAválpas
    ELSE
      IF COUNTER <> 2
        STORE counter+1 TO counter
        ERASE
        @ 12,25 SAY " INVALID PASSWORD "
        @ 14,25 SAY " CHECK YOUR PASSWORD"
        DO pause
      ELSE
        STORE COUNTER+1 TO COUNTER
      ENDIF
    ENDIF
  ENDIF
ENDIF 1

IF counter = 3
  STCRE t TO thru
  ERASE
  @ 12,20 SAY "*****"
  @ 13,20 SAY "***** TERMINATE ACCESS *****"
  @ 14,20 SAY "*****"
ENDIF

ENDDC
RELEASE counter, pause, passrec, validid, THRU
RETURN

```


APPENDIX E COMMAND DETERMINER MODULE LISTING

A. CCMMAND DETERMINER ROUTINE

```

* Routine Name: Cmddet.prg
* Module Name: Command Determiner Module
* Version: 3.0
* Author: D.P. Haeusler
* Date: 30 Oct 83
* Variables Used: aafinish, cmdopt, helpopt
* Variables Modified: aafinish, cmdopt, helpopt
* Variables Created: aafinish, cmdopt, helpopt
* Variables Released: cmdopt, aafinish, helpopt
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Master
* Description: This routine creates a menu which is used
* to determine which options of the program will be
* executed by the user. These options allow the user to
* control the use of the DBMS by the program.

```

STORE f TO AAfinish

DO WHILE .NOT. AAfinish

```

ERASE
@ 2,17 SAY "                USMC INDIVIDUAL TRAINING      "
@ 3,17 SAY "                RECORD SYSTEM              "
@ 5,17 SAY "                MAIN MENU                    "
@ 7,17 SAY "    SELECT ONE OF THE FOLLOWING OPTIONS BY"
@ 8,17 SAY "    ENTERING THE APPROPRIATE LETTER      "
@ 10,17 SAY "    D...DATA ENTRY AND SYSTEM FUNCTIONS"
@ 11,17 SAY "    S...STANDARD REPORTS                  "
@ 12,17 SAY "    N...NCNSTANDARD REPORTS               "
@ 13,17 SAY "    H...HELP                              "
@ 14,17 SAY "    Q...QUIT                              "

```

ACCEPT " SELECT OPTION ====>" TO cmdopt

```

DC CASE
CASE !(cmdopt) = 'D'
    DO maintain
    STORE f TO AAfinish
CASE !(cmdopt) = 'S'
    DO STDRPT
    STORE f TO AAfinish
CASE !(cmdopt) = 'N'
    DO nstdtde
    STORE f TO AAfinish
CASE !(cmdopt) = 'H'
    STORE 1 TO HELPOPT
    DO B:HELP2
    STORE f TO AAfinish
CASE !(cmdopt) = 'Q'
    STORE t TO AAfinish
ENDCASE

```

```

ENDDC
RELEASE cmdopt, HELPOPT, aafinish
RETURN

```


E. HELP2 ROUTINE

```
* Routine Name: Help2.prg
* Module Name: Maintenance Module
* Version: 3.0.1
* Author: D.P. Haeusler
* Date: 6.0.1.0
* Variables Used: helpopt
* Variables Modified: none
* Variables Created: none
* Variables Released: none
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Maintain.prg, Cmddet.prg
* Description: This routine gives a brief description of
* the commands that can be invoked by the user.
```

```
ERASE
TO CASE
CASE HELFOPT = 1
TEXT
```

D...DATA ENTRY AND SYSTEM FUNCTIONS: This module allows the user to enter data on individual Marines into the data base and to modify data for system functions. Installation occurs in this module and any manipulation of the data such as corrections and updates of the information.

S...STANDARD REPORTS: This module produces reports that are used on a day in and day out basis. The reports are displayed on the screen and if a paper copy is required while the data is still on the screen, press the shift key and the prtsc key at the same time, this will print everything on the screen.

N...NONSTANDARD REPORTS: This module allows the users to construct their own queries of the data base. It prompts you for the information to make a query. Paper copies of the reports can be obtained by the same method as in standard reports.

```
ENDTEXT
@ 23,15 SAY " **** PRESS ANY KEY TO CONTINUE ****"
SET CONSOLE OFF
WAIT
SET CONSOLE ON
CASE HELFOPT = 2
TEXT
```

V...VIEW INDIVIDUAL DATA: This module prompts you for the name of an individual whose ITR you wish to see. This module does not allow you to change any fields all you can do is scan the data.

U...UPDATE DATA: This module allows you to change or enter data in any field of any ITR. The ITR must already exist and is identified by the individual's name. You only need to enter enough of the name to be able to distinguish it from another Marine's name.

C...CREATE AN ITR: This module allows the creation of new ITR's. It will prompt you to enter all the appropriate data. Enter the data in one field at a time. If no data is to be entered in that field, strike the enter key leaving the field empty.

D...DELETE AN ITR: This module allows the Data Base Administrator to delete records in the data base. Once a

record has been deleted it is gone forever and can not be recovered. Access to this module is restricted.

S...SYSTEM FUNCTION: This module allows the housekeeping of the system to be accomplished by the Data Base Administrator. Access to this module is restricted.

ENDTEXT

@ 23.15 SAY " **** PRESS ANY KEY TO CONTINUE ****"

SET CONSOLE OFF

WAIT

SET CONSOLE ON

ENDCASE

RETURN

APPENDIX F

NONSTANDARD COMMAND GENERATOR MODULE LISTING

A. NONSTANDARD REPORT ROUTINE

```

* ROUTINE NAME: NOSTDEPT.PRG
* MODULE NAME: NONSTANDARD COMMAND GENERATOR
* VERSION: 4.0
* AUTHOR: R.E. PRUIETT
* DATE: 04FEB84
* VARIABLES USED: MATRI1, DESC1, MATRI2, DESC2, MATRI3,
* DESC3, SLCT1, SLCT2, SLCT3
* VARIABLES MODIFIED: MREPEAT, COUNT, PROCEED
* VARIABLES CREATED: MREPEAT, COUNT, PROCEED
* VARIABLES RELEASED: M??????, REPEAT, COUNT,
* ATRI????, SLCT????, DESC????
* FILES OPENED/CLOSED: A:PERS, E:EST, B:QUAL
* TEMP FILES CREATED: TEMP, SLCTDATA, TEMP2, TEMP3
* USING SUBROUTINES: CCOMMAND DETERMINER MODULE
* DESCRIPTION: THIS MODULE SEARCHES THE APPROPRIATE
* DATABASE FILES IN ORDER TO LOCATE AND DISPLAY ALL RECORDS
* THAT SATISFY A SET OF FROM ONE TO THREE USER SPECIFIED
* ATTRIBUTES.
*
STORE T TO MREPEAT
DO WHILE MREPEAT = T
  STORE F TO MREPEAT
  ERASE
  @ 10,25 SAY "NONSTANDARD COMMAND GENERATOR"
  @ 20,0 SAY "PRESS 'H' FOR HELP"
  @ 21,0 SAY "      CF "
  ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
  IF ! (PROCEED) = "H"
    DO A:NCSTDHLP
    STORE T TO MREPEAT
  ENDIF
ENDDO
ERASE
@ 1,25 SAY "NONSTANDARD COMMAND GENERATOR"
@ 3,5 SAY "WHEN PROMPTED ENTER AN ATTRIBUTE AND A";
"DESCRIPTOR FOR EACH"
@ 4,5 SAY "CONDITION STATEMENT DESIRED.      "
@ 6,10 SAY "LOCATE ALL MARINES THAT SATISFY THESE;
CONDITIONS:"
@ 8,10 SAY "      <ATTRIBUTE>                <DESCRIPTOR>"
@ 9,10 SAY "      -----                IS                -----"
@ 20,0
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED

STORE T TO MREPEAT
STORE 0 TO COUNT
DO WHILE MREPEAT = T .AND. COUNT <= 3
  STORE COUNT + 1 TO COUNT
  STORE F TO MREPEAT
  DO SLCTATRI
  ERASE
  DO CASE
    CASE COUNT = 1
      @ 1,0 SAY "<ATTRIBUTE>                <DESCRIPTOR>"
      @ 2,0 SAY "      -----                -----"
      @ 4,6 SAY MATRI1-----

```



```

@ 4,30 SAY "IS"
@ 4,48 SAY DESC1
CASE COUNT = 2
@ 1,0 SAY "<ATTRIBUTE>" <DESCRIPTION> "
@ 2,0 SAY "
@ 4,6 SAY MATRI1 @ 4,30 SAY "IS"-----
@ 4,48 SAY DESC1
@ 6,30 SAY "AND"
@ 8,6 SAY MATRI2
@ 8,30 SAY "IS" @ 8,48 SAY DESC2 CASE COUNT = 3
@ 1,0 SAY "<ATTRIBUTE>" <DESCRIPTION> "
@ 2,0 SAY "-----"
@ 4,6 SAY MATRI1-----
@ 4,30 SAY "IS"
@ 4,48 SAY DESC1
@ 6,30 SAY "AND"
@ 8,6 SAY MATRI2
@ 8,30 SAY "IS"
@ 8,48 SAY DESC2
@ 10,30 SAY "AND"
@ 12,6 SAY MATRI3
@ 12,30 SAY "IS"
@ 12,48 SAY DESC3
ENDIF
IF COUNT < 3
@ 20,0 SAY "DO YOU WANT TO INCLUDE ANOTHER CONDITION";
"STATEMENT?"
ACCEPT "(Y/N) ==>" TO PROCEED
IF PROCEED = "Y"
SECRET TO REPEAT
ENDIF
ENDIF
IF PROCEED = "N" .CR. COUNT >= 3
@ 22,0
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDIF
ENDDC

ERASE
IF COUNT = 1
DO CASE
CASE (SLCT1 >= "A" .AND. SLCT1 <= "I")
USE A:PERS
DISPLAY ALL FOR &ATRI1 = "&DESC1" OFF FIELDS;
NAME,SSN,&ATRI1
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J" .OR. SLCT1 = "K")
USE B:EST
CCPY TO A:TEMP FIELD SSN,&ATRI1 FOR &ATRI1 =;
"&DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERS
JCIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
NAME,SSN,&ATRI1
USE SLCTDATA
DISPLAY ALL
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 >= "L" .AND. SLCT1 <= "N")
USE B:QUAL
CCPY TO A:TEMP FIELD SSN,&ATRI1 FOR &ATRI1 =;
"&DESC1"
SELECT PRIMARY
USE A:TEMP

```



```

SELECT SECONDARY
USE A:PERS
JCIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
NAME,SSN,&ATRI1
USE SLCTDATA
DISPLAY ALL
IF EOF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDIF

IF CCUNT = 2
DO CASE
CASE {SLCT1 >= "A" .AND. SLCT1 <= "I"} .AND.;
      {SLCT2 >= "A" .AND. SLCT2 <= "I"}
      USE A:PERS
      COPY TO A:TEMP FIELD NAME,SSN,&ATRI1,&ATRI2 FCR;
      &ATRI1="&DESC1"
      USE A:TEMP
      DISPLAY FOR &ATRI2="&DESC2" OFF FIELDS;
      NAME,SSN,&ATRI1,&ATRI2
      IF EOF
        @ 23,50 SAY "SEARCH COMPLETE"
      ENDIF
CASE (SLCT1 >= "A" .AND. SLCT1 <= "I") .AND.;
      (SLCT2 = "J" .OR. SLCT2 = "K")
      USE B:EST
      COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2 =;
      "&DESC2"
      SELECT PRIMARY
      USE A:TEMP
      SELECT SECONDARY
      USE A:PERS
      JCIN TO SLCTDATA FCR P.SSN=S.SSN FIELDS;
      NAME,SSN,&ATRI1,&ATRI2
      USE SLCTDATA
      DISPLAY FOR &ATRI1 = "&DESC1" OFF FIELDS;
      NAME,SSN,&ATRI1,&ATRI2
      IF EOF
        @ 23,50 SAY "SEARCH COMPLETE"
      ENDIF
CASE (SLCT1 >= "A" .AND. SLCT1 <= "I") .AND.;
      (SLCT2 >= "I" .AND. SLCT2 <= "N")
      USE B:QUAL
      COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2 =;
      "&DESC2"
      SELECT PRIMARY
      USE A:TEMP
      SELECT SECONDARY
      USE A:PERS
      JCIN TO SLCTDATA FCR P.SSN=S.SSN FIELDS;
      NAME,SSN,&ATRI1,&ATRI2
      USE SLCTDATA
      DISPLAY FOR &ATRI1 = "&DESC1" OFF FIELDS;
      NAME,SSN,&ATRI1,&ATRI2
      IF EOF
        @ 23,50 SAY "SEARCH COMPLETE"
      ENDIF
CASE (SLCT1 = "J" .OR. SLCT1 = "K") .AND.;
      (SLCT2 >= "A" .AND. SLCT2 <= "I")
      USE A:PERS
      COPY TO A:TEMP FIELD NAME,SSN,&ATRI2 FOR &ATRI2 =;
      "&DESC2"
      SELECT PRIMARY
      USE A:TEMP
      SELECT SECONDARY
      USE B:EST

```



```

JCIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
NAME,SSN,&ATRI1,&ATRI2
USE SLCTDATA
DISPLAY FOR &ATRI1 = "&DESC1" OFF FIELDS;
NAME,SSN,&ATRI1,&ATRI2
IF EOF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J" .OR. SLCT1 = "K") .AND.;
(SLCT2 = "J" .OR. SLCT2 = "K")
USE B:EST
COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI2 FOR &ATRI2;
= "&DESC2" .AND. &ATRI1 = "&DESC1"
SELECT PRIMAFY
USE A:TEMP
SELECT SECONIARY
USE A:PERS
JCIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
NAME,SSN,&ATRI1,&ATRI2
USE SLCTDATA
DISPLAY ALL
IF EOF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 = "J" .OR. SLCT1 = "K") .AND.;
(SLCT2 >= "I" .AND. SLCT2 <= "N")
USE B:QUAL
COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2 =;
"&DESC2"
USE B:EST
COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR &ATRI1 =;
"DESC1"
SELECT PRIMAFY
USE A:TEMP
SELECT SECONIARY
USE A:TEMP2
JCIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS;
SSN,&ATRI1,&ATRI2
SELECT PRIMAFY
USE A:TEMP3
SELECT SECONIARY
USE A:PERS
JCIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
NAME,SSN,&ATRI1,&ATRI2
USE SLCTDATA
DISPLAY ALL
IF EOF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 >= "I" .AND. SLCT1 <= "N") .AND.;
(SLCT2 >= "A" .AND. SLCT2 <= "I")
USE A:PERS
COPY TO A:TEMP FIELD NAME,SSN,&ATRI2 FOR &ATRI2 =;
"&DESC2"
SELECT PRIMAFY
USE A:TEMP
SELECT SECONIARY
USE B:QUAL
JCIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
NAME,SSN,&ATRI1,&ATRI2
USE SLCTDATA
DISPLAY FOR &ATRI1 = "&DESC1" OFF FIELDS;
NAME,SSN,&ATRI1,&ATRI2
IF EOF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 >= "I" .AND. SLCT1 <= "N") .AND.;
(SLCT2 = "J" .OR. SLCT2 = "K")
USE B:EST

```



```

COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2 =;
"&DESC2"
USE B:QUAL
COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR &ATRI1 =;
"DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO A:TEMP3 FOR F.SSN=S.SSN FIELDS;
SSN,&ATRI1,&ATRI2
SELECT PRIMARY
USE A:TEMP3
SELECT SECONDARY
USE A:PERS
JOIN TO SLCTDATA FOR F.SSN=S.SSN FIELDS;
NAME,SSN,&ATRI1,&ATRI2
USE SLCTDATA
DISPLAY ALL
IF EOF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT1 >= "I" .AND. SLCT1 <= "N") .AND.;
(SLCT2 >= "L" .AND. SLCT2 <= "N")
USE B:QUAL
COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI2 FOR &ATRI1;
=&DESC1" .AND. &ATRI2 = "&DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERS
JOIN TO SLCTDATA FOR F.SSN=S.SSN FIELDS;
NAME,SSN,&ATRI1,&ATRI2
USE SLCTDATA
DISPLAY ALL
IF EOF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENICASE
ACCEPT "PRESS 'ENTER' TO CCNTINUE" TO PROCEED
ENDIF
IF CCNT = 3
DO CASE
  CASE (SLCT1 >= "A" .AND. SLCT1 <= "I")
  DO CASE
    CASE (SLCT2 >= "A" .AND. SLCT2 <= "I")
    DO CASE
      CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
      USE A:PERS
      COPY TO A:TEMP FIELD NAME, SSN, &ATRI1,;
      &ATRI2, &ATRI3 FOR &ATRI1="&DESC1"
      USE A:TEMP
      COPY TO A:TEMP2 FIELD NAME, SSN, &ATRI1,;
      &ATRI2, &ATRI3 FOR &ATRI2="&DESC2"
      USE A:TEMP2
      DISPLAY FOR &ATRI3="&DESC3" OFF FIELDS;
      NAME, SSN, &ATRI1, &ATRI2, &ATRI3
      IF EOF
        @ 23,50 SAY "SEARCH COMPLETE"
      ENDIF
      CASE (SLCT3 = "J" .OR. SLCT3 = "K")
      USE F:EST
      COPY TO A:TEMP FIELD SSN,&ATRI3 FOR &ATRI3;
      = "&DESC3"
      SELECT PRIMARY
      USE A:TEMP
      SELECT SECONDARY
      USE A:PERS

```



```

        JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
        NAME,SSN,&ATRI1, &ATRI2,&ATRI3
        USE SICTDATA
        DISPLAY FOR &ATRI1 = "&DESC1" .AND. &ATRI2;
        = "&DESC2" OFF FIELDS NAME, SSN, &ATRI1,;
        &ATRI2, &ATRI3
        IF ECF
            @ 23,50 SAY "SEARCH COMPLETE"
        ENDIF
    CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
        USE E:QUAL
        COPY TO A:TEMP FIELD SSN,&ATRI3 FOR &ATRI3;
        = "&DESC3"
        SELECT PRIMARY
        USE A:TEMP
        SELECT SECONDARY
        USE A:PERS
        JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
        NAME, SSN, &ATRI1, &ATRI2, &ATRI3
        USE SICTDATA
        DISPLAY FOR &ATRI1 = "&DESC1" .AND. &ATRI2;
        = "&DESC2" OFF FIELDS NAME, SSN, &ATRI1,;
        &ATRI2, &ATRI3
        IF ECF
            @ 23,50 SAY "SEARCH COMPLETE"
        ENDIF
    ENDCASE
CASE (SLCT2 >= "L" .AND. SLCT2 <= "N")
    DO CASE
        CASE (SICT3 >= "A" .AND. SLCT3 <= "I")
            USE E:QUAL
            COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2;
            = "&DESC2"
            SELECT PRIMARY
            USE A:TEMP
            SELECT SECONDARY
            USE A:PERS
            JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
            NAME, SSN, &ATRI1, &ATRI2, &ATRI3
            USE SICTDATA
            DISPLAY FOR &ATRI1 = "&DESC1" .AND. &ATRI3;
            = "&DESC3" OFF FIELDS NAME, SSN, &ATRI1,;
            &ATRI2, &ATRI3
            IF ECF
                @ 23,50 SAY "SEARCH COMPLETE"
            ENDIF
        CASE (SICT3 >= "I" .AND. SLCT3 <= "N")
            USE A:PERS
            COPY TO A:TEMP FIELD NAME,SSN,&ATRI1 FOR;
            &ATRI1 = "&DESC1"
            SELECT PRIMARY
            USE A:TEMP
            SELECT SECONDARY
            USE E:QUAL
            JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
            NAME, SSN, &ATRI1, &ATRI2, &ATRI3
            USE SICTDATA
            DISPLAY FOR &ATRI2 = "&DESC2" .AND. &ATRI3;
            = "&DESC3" OFF FIELDS NAME, SSN, &ATRI1,;
            &ATRI2, &ATRI3
            IF ECF
                @ 23,50 SAY "SEARCH COMPLETE"
            ENDIF
        CASE (SICT3 = "J" .OR. SLCT3 = "K")
            USE A:PERS
            COPY TO A:TEMP FIELD NAME,SSN,&ATRI1 FOR;
            &ATRI1 = "&DESC1"
            SELECT PRIMARY
            USE A:TEMP

```



```

SELECT SECONDARY
USE E:QUAL
JOIN TO A:TEMP2 FOR P.SSN=S.SSN FIELDS;
NAME, SSN, &ATRI1, &ATRI2
SELECT PRIMARY
USE A:TEMP2
SELECT SECONDARY
USE E:EST
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
NAME, SSN, &ATRI1, &ATRI2,&ATRI3
USE SLCTDATA
DISPLAY FOR &ATRI2 = "&DESC2" .AND. &ATRI3;
= "&DESC3" OFF FIELDS NAME, SSN, &ATRI1,;
&ATRI2, &ATRI3
IF ECF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
DO CASE
CASE (SLCT2 = "J" .OR. SLCT2 = "K")
  CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
    USE E:EST
    COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2;
    = "&DESC2"
    SELECT PRIMARY
    USE A:TEMP
    SELECT SECONDARY
    USE A:PERS
    JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
    NAME, SSN, &ATRI1, &ATRI2, &ATRI3
    USE SLCTDATA
    DISPLAY FOR &ATRI1 = "&DESC1" .AND. &ATRI3;
    = "&DESC3" OFF FIELDS NAME, SSN, &ATRI1,;
    &ATRI2, &ATRI3
    IF ECF
      @ 23,50 SAY "SEARCH COMPLETE"
    ENDIF
  CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
    USE A:PERS
    COPY TO A:TEMP FIELD NAME,SSN,&ATRI1 FOR;
    &ATRI1 = "&DESC1"
    USE E:EST
    COPY TO A:TEMP2 FIELD SSN,&ATRI2 FOR;
    &ATRI2 = "&DESC2"
    SELECT PRIMARY
    USE A:TEMP
    SELECT SECONDARY
    USE A:TEMP2
    JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS;
    NAME, SSN, &ATRI1, &ATRI2,&ATRI3
    SELECT PRIMARY
    USE A:TEMP3
    SELECT SECONDARY
    USE E:QUAL
    JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
    NAME, SSN, &ATRI1, &ATRI2, &ATRI3
    USE SLCTDATA
    DISPLAY FOR &ATRI3 = "&DESC3" OFF FIELDS;
    NAME, SSN, &ATRI1, &ATRI2, &ATRI3
    IF ECF
      @ 23,50 SAY "SEARCH COMPLETE"
    ENDIF
  CASE (SLCT3 = "J" .OR. SLCT3 = "K")
    USE A:PERS
    COPY TO A:TEMP FIELD NAME,SSN,&ATRI1 FOR;
    &ATRI1 = "&DESC1"
    SELECT PRIMARY
    USE A:TEMP
    SELECT SECONDARY

```



```

        USE E:EST
        JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
        NAME, SSN, &ATRI1, &ATRI2, &ATRI3
        USE SLCTDATA
        DISPLAY FOR &ATRI2 = "&DESC2" .AND. &ATRI3;
        = "&DESC3" OFF FIELDS NAME, SSN, &ATRI1,;
        &ATRI2, &ATRI3
        IF ECF
            @ 23,50 SAY "SEARCH COMPLETE"
        ENDIF
    ENDCASE
ENDCASE
**BEGIN QUAL / * / *
CASE (SLCT1 >= "I" .AND. SLCT1 <= "N")
    **BEGIN QUAL / PERS / *
    DO CASE
        CASE (SLCT2 >= "A" .AND. SLCT2 <= "I")
            DO CASE
                CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
                    USE A:PERS
                    COPY TO A:TEMP FIELD NAME, SSN, &ATRI2,;
                    &ATRI3 FOR &ATRI2="&DESC2" .AND.;
                    &ATRI3="&DESC3"
                    SELECT PRIMARY
                    USE A:TEMP
                    SELECT SECONDARY
                    USE E:QUAL
                    JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
                    NAME, SSN, &ATRI1, &ATRI2, &ATRI3
                    USE SLCTDATA
                    DISPLAY FOR &ATRI1="&DESC1" OFF FIELDS;
                    NAME, SSN, &ATRI1, &ATRI2, &ATRI3
                    IF ECF
                        @ 23,50 SAY "SEARCH COMPLETE"
                    ENDIF
                CASE (SLCT3 = "J" .OR. SLCT3 = "K")
                    USE E:EST
                    COPY TO A:TEMP FIELD SSN,&ATRI3 FOR &ATRI3;
                    = "&DESC3"
                    USE E:QUAL
                    COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
                    &ATRI1="&DESC1"
                    SELECT PRIMARY
                    USE A:TEMP
                    SELECT SECONDARY
                    USE A:TEMP2
                    JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS;
                    SSN, &ATRI1, &ATRI3
                    SELECT PRIMARY
                    USE A:TEMP3
                    SELECT SECONDARY
                    USE A:PERS
                    JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
                    NAME, SSN, &ATRI1, &ATRI2, &ATRI3
                    USE SLCTDATA
                    DISPLAY FOR &ATRI2 = "&DESC2";
                    OFF FIELDS NAME,SSN,&ATRI1,&ATRI2,&ATRI3
                    IF ECF
                        @ 23,50 SAY "SEARCH COMPLETE"
                    ENDIF
                CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
                    USE E:QUAL
                    COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI3;
                    FOR &ATRI3 = "&DESC3" .AND. &ATRI1 =;
                    "&DESC1"
                    SELECT PRIMARY
                    USE A:TEMP
                    SELECT SECONDARY
                    USE A:PERS

```



```

        JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
        NAME, SSN, &ATRI1, &ATRI2, &ATRI3
        USE SICTDATA
        DISPLAY FOR &ATRI2 = "&DESC2";
        OFF FIELDS NAME,SSN,&ATRI1,&ATRI2,&ATRI3
        IF EOF
            @ 23,50 SAY "SEARCH COMPLETE"
        ENDIF
    ENDCASE
CASE (SLCT2 >= "L" .AND. SLCT2 <= "N")
    DO CASE
        CASE (SICT3 >= "A" .AND. SLCT3 <= "I")
            USE E:QUAL
            COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI2;
            FOR &ATRI1 = "&DESC1" .AND. &ATRI2 =;
            "&DESC2"
            SELECT PRIMARY
            USE A:TEMP
            SELECT SECONDARY
            USE A:PERS
            JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
            NAME, SSN, &ATRI1, &ATRI2, &ATRI3
            USE SICTDATA
            DISPLAY FOR &ATRI3 = "&DESC3";
            OFF FIELDS NAME,SSN,&ATRI1,&ATRI2,&ATRI3
            IF EOF
                @ 23,50 SAY "SEARCH COMPLETE"
            ENDIF
        CASE (SICT3 >= "I" .AND. SLCT3 <= "N")
            USE E:QUAL
            COPY TO A:TEMP FIELD SSN, &ATRI1, &ATRI2,;
            &ATRI3 FOR &ATRI1 = "&DESC1" .AND. &ATRI2;
            = "&DESC2" .AND. &ATRI3 = "&DESC3"
            SELECT PRIMARY
            USE A:TEMP
            SELECT SECONDARY
            USE A:PERS
            JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
            NAME, SSN, &ATRI1, &ATRI2, &ATRI3
            USE SICTDATA
            DISPLAY ALL OFF FIELDS NAME,SSN,;
            &ATRI1,&ATRI2,&ATRI3
            IF EOF
                @ 23,50 SAY "SEARCH COMPLETE"
            ENDIF
        CASE (SLCT3 = "J" .OR. SLCT3 = "K")
            USE E:QUAL
            COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI2;
            FOR &ATRI1 = "&DESC1" .AND. &ATRI2 =;
            "&DESC2"
            USE E:EST
            COPY TO A:TEMP2 FIELD SSN,&ATRI3 FOR;
            &ATRI3 = "&DESC3"
            SELECT PRIMARY
            USE A:TEMP
            SELECT SECONDARY
            USE A:TEMP2
            JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS;
            SSN, &ATRI1, &ATRI2, &ATRI3
            SELECT PRIMARY
            USE A:TEMP3
            SELECT SECONDARY
            USE A:PERS
            DISPLAY FOR P.SSN=S.SSN OFF FIELDS NAME,;
            SSN, &ATRI1, &ATRI2, &ATRI3.
            IF EOF
                @ 23,50 SAY "SEARCH COMPLETE"
            ENDIF
    ENDCASE

```



```

CASE (SLCT2 = "J" .OR. SLCT2 = "K")
  IO CASE
    CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
      USE E:EST
      COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2;
      = "&DESC2"
      USE E:QUAL
      COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
      &ATRI1 = "&DESC1"
      SELECT PRIMARY
      USE A:TEMP
      SELECT SECONDARY
      USE A:TEMP2
      JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS;
      SSN,&ATRI1,&ATRI2
      SELECT PRIMARY
      USE A:TEMP3
      SELECT SECONDARY
      USE A:PERS
      DISPLAY FOR S.&ATRI3="&DESC3" .AND.;
      P.SSN=S.SSN OFF FIELDS NAME, SSN, &ATRI1,;
      &ATRI2, &ATRI3
      IF ECF
        @ 23,50 SAY "SEARCH COMPLETE"
      ENDIF
    CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
      USE E:QUAL
      COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI3;
      FOR &ATRI1 = "&DESC1" .AND. &ATRI3 =;
      "&DESC3"
      USE E:EST
      COPY TO A:TEMP2 FIELD SSN,&ATRI2 FOR;
      &ATRI2 = "&DESC2"
      SELECT PRIMARY
      USE A:TEMP
      SELECT SECONDARY
      USE A:TEMP2
      JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS;
      SSN, &ATRI1, &ATRI2, &ATRI3
      SELECT PRIMARY
      USE A:TEMP3
      SELECT SECONDARY
      USE A:PERS
      DISPLAY FOR P.SSN=S.SSN OFF FIELDS NAME,;
      SSN, &ATRI1, &ATRI2, &ATRI3
      IF ECF
        @ 23,50 SAY "SEARCH COMPLETE"
      ENDIF
    CASE (SLCT3 = "J" .OR. SLCT3 = "K")
      USE E:EST
      COPY TO A:TEMP FIELD SSN,&ATRI2,&ATRI3;
      FOR &ATRI2 = "&DESC2" .AND. &ATRI3 = "&DESC3"
      USE E:QUAL
      COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
      &ATRI1 = "&DESC1"
      SELECT PRIMARY
      USE A:TEMP
      SELECT SECONDARY
      USE A:TEMP2
      JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS;
      SSN, &ATRI1, &ATRI2, &ATRI3
      SELECT PRIMARY
      USE A:TEMP3
      SELECT SECONDARY
      USE A:PERS
      DISPLAY FOR P.SSN=S.SSN;
      OFF FIELDS NAME,SSN,&ATRI1,&ATRI2,&ATRI3
      IF ECF
        @ 23,50 SAY "SEARCH COMPLETE"

```



```

        ENDIF
    ENDCASE
  ENICASE
** BEGIN EST / * / *
CASE (SLCT1 = "J" .OR. SLCT1 = "K")
  **BEGIN EST / PERS / *
  DO CASE
    CASE (SLCT2 >= "A" .AND. SLCT2 <= "I")
      DO CASE
        CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
          USE A:PERS
          COPY TO A:TEMP FIELD NAME, SSN, &ATRI2,;
          &ATRI3 FOR &ATRI2 = "&DESC2" .AND. &ATRI3;
          = "&DESC3"
          USE E:EST
          COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
          &ATRI1="&DESC1"
          SELECT PRIMARY
          USE A:TEMP
          SELECT SECONDARY
          USE A:TEMP2
          DISPLAY FOR P.SSN=S.SSN OFF FIELDS NAME,;
          SSN, &ATRI1, &ATRI2, &ATRI3
          IF ECF
            @ 23,50 SAY "SEARCH COMPLETE"
          ENDIF
        CASE (SLCT3 = "J" .OR. SLCT3 = "K")
          USE E:EST
          COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI3;
          FOR &ATRI1="&DESC1" .AND. &ATRI3="&DESC3"
          SELECT PRIMARY
          USE A:TEMP
          SELECT SECONDARY
          USE A:PERS
          DISPLAY FOR P.SSN=S.SSN .AND. S.&ATRI2 =;
          "&DESC2" OFF FIELDS NAME, SSN, &ATRI1,;
          &ATRI2, &ATRI3
          IF ECF
            @ 23,50 SAY "SEARCH COMPLETE"
          ENDIF
        CASE (SLCT3 >= "L" .AND. SLCT3 <= "N")
          USE E:QUAL
          COPY TO A:TEMP FIELD SSN,&ATRI3 FOR &ATRI3;
          = "&DESC3"
          USE E:EST
          COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
          &ATRI1="&DESC1"
          SELECT PRIMARY
          USE A:TEMP
          SELECT SECONDARY
          USE A:TEMP2
          JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
          SSN, &ATRI1, &ATRI3
          SELECT PRIMARY
          USE SLCTDATA
          SELECT SECONDARY
          USE A:PERS
          DISPLAY FOR P.SSN=S.SSN .AND. &ATRI2 =;
          "&DESC2" OFF FIELDS NAME, SSN, &ATRI1,;
          &ATRI2, &ATRI3
          IF ECF
            @ 23,50 SAY "SEARCH COMPLETE"
          ENDIF
        ENDCASE
      CASE (SLCT2 >= "L" .AND. SLCT2 <= "N")
        DO CASE
          CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
            USE E:QUAL
            COPY TO A:TEMP FIELD SSN,&ATRI2 FOR &ATRI2;

```



```

= "&DESC2"
USE E:EST
COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
&ATRI1="&DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDAFY
USE A:TEMP2
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
SSN, &ATRI1, &ATRI2
SELECT PRIMARY
USE SLCTDATA
SELECT SECONDAFY
USE A:PERS
DISPIAY FOR &ATRI3 = "&DESC3" .AND.;
P.SSN=S.SSN OFF FIELDS NAME, SSN, &ATRI1,;
&ATRI2, &ATRI3
IF ECF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 >= "I" .AND. SLCT3 <= "N")
USE E:QUAL
COPY TO A:TEMP FIELD SSN,&ATRI2,&ATRI3;
FOR &ATRI2="&DESC2" .AND. &ATRI3="&DESC3"
USE E:EST
COPY TO A:TEMP2 FIELD SSN,&ATRI1 FOR;
&ATRI1="&DESC1"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDAFY
USE A:TEMP2
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE SLCTDATA
SELECT SECONDAFY
USE A:PERS
DISPIAY FOR P.SSN=S.SSN OFF FIELDS NAME,;
SSN, &ATRI1, &ATRI2, &ATRI3
IF ECF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (SLCT3 = "J" .OR. SLCT3 = "K")
USE E:EST
COPY TO A:TEMP FIELD SSN,&ATRI1,&ATRI3;
FOR &ATRI1="&DESC1" .AND. &ATRI3="&DESC3"
USE E:QUAL
COPY TO A:TEMP2 FIELD SSN,&ATRI2 FOR;
&ATRI2="&DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDAFY
USE A:TEMP2
JOIN TO A:TEMP3 FOR P.SSN=S.SSN FIELDS;
SSN, &ATRI1, &ATRI2, &ATRI3
SELECT PRIMARY
USE A:TEMP3
SELECT SECONDAFY
USE A:PERS
DISPIAY FOR P.SSN=S.SSN OFF FIELDS NAME,;
SSN, &ATRI1, &ATRI2, &ATRI3
IF ECF
  @ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
CASE (SLCT2 = "J" .CR. SLCT2 = "K")
  IO CASE
    CASE (SLCT3 >= "A" .AND. SLCT3 <= "I")
      USE E:EST

```



```

COPY TO A:TEMP FIELD SSN,&A1R11,&A1R12 ;
FOR &A1R11="&DESC1" .AND. &A1R12="&DESC2"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERS
DISP1AY FOR P.SSN=S.SSN .AND. &A1R13 =;
"&DESC3" OFF FIELDS NAME, SSN, &A1R11,;
&A1R12, &A1R13
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (S1CT3 >= "L" .AND. SLCT3 <= "N")
USE E:EST
COPY TO A:TEMP FIELD SSN,&A1R11,&A1R12;
FOR &A1R11="&DESC1" .AND. &A1R12="&DESC2"
USE E:QUAL
COPY TO A:TEMP2 FIELD SSN,&A1R13 FOR;
&A1R13 = "&DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:TEMP2
JOIN TO SLCTDATA FOR P.SSN=S.SSN FIELDS;
SSN, &A1R11, &A1R12, &A1R13
SELECT PRIMARY
USE SLCTDATA
SELECT SECONDARY
USE A:PERS
DISP1AY FOR P.SSN=S.SSN OFF FIELDS NAME,;
SSN, &A1R11, &A1R12, &A1R13
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
CASE (S1CT3 = "J" .OR. SLCT3 = "K")
USE A:EST
COPY TO A:TEMP FIELD SSN, &A1R11, &A1R12,;
&A1R13 FOR &A1R11 = "&DESC1" .AND. &A1R12,;
= "&DESC2" .AND. &A1R13 = "&DESC3"
SELECT PRIMARY
USE A:TEMP
SELECT SECONDARY
USE A:PERS
DISP1AY FOR P.SSN=S.SSN OFF FIELDS NAME,;
SSN, &A1R11, &A1R12, &A1R13
IF ECF
@ 23,50 SAY "SEARCH COMPLETE"
ENDIF
ENDCASE
ENDCASE
ENDCASE
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDIF
USE
RELEASE ALL LIKE M???????
RELEASE REPEAT,COUNT
RELEASE ALL LIKE A1R1????
RELEASE ALL LIKE SLCT????
RELEASE ALL LIKE DESC????
SET CCNSCLE OFF
DELETE FILE A:TEMP
DELETE FILE A:TEMP2
DELETE FILE A:TEMP3
SET CCNSCLE ON
RETURN

```


E. NCNSTANDARD HELP ROUTINE

```
* ROUTINE NAME: NOSTDHL.PRG
* MODULE NAME: NCNSTANDARD REPCRT GENERATOR
* VERSION: 4.1.0
* AUTHCR: R.E. PRUIETT
* DATE: 10FEB84
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: PROCEED
* VARIABLES RELEASED: NONE
* FILES CPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: NCSTDRT.PRG (4.0)
* DESCRIPTION: THIS ROUTINE SUPPLIES RESPONSE TO THE USER'S
* REQUEST FOR "HELP" WHILE IN THE NONSTANDARD REPORT MENU.
```

```
ERASE
@ 0,25 SAY "NONSTANDARD COMMAND GENERATOR"
@ 2,0 SAY "LOCATE ALL MARINES THAT SATISFY THE FOLLOWING;
CONDITIONS:"
@ 4,0 SAY " <ATTRIBUTE> <DESCRIPTOR>"
@ 4,0 SAY " -----"
@ 5,0 SAY " "
@ 6,0 SAY " "
@ 7,0 SAY " RANK IS E5 "
@ 9,0 SAY " -----"
@ 10,0 SAY " "
@ 11,0 SAY " UP TO THREE CONDITION STATEMENTS MAY BE;
JOINED "
@ 12,0 SAY " FOR EXAMPLE "
@ 13,0 SAY "LOCATE ALL MARINES THAT SATISFY THE FOLLOWING;
CONDITIONS:"
@ 14,0 SAY " <ATTRIBUTE> <DESCRIPTOR>"
@ 15,0 SAY " "
@ 16,0 SAY " RANK IS E5 "
@ 17,0 SAY " AND "
@ 18,0 SAY " COMPANY IS A CO "
@ 19,0 SAY " AND "
@ 20,0 SAY " PRIMARY MOS IS 0331 "
@ 21,0 SAY "JOINING THESE THREE CONDITION STATEMENTS GIVE;
YOU THE NAME "
@ 22,0 SAY "AND SSN OF ALL E5'S IN A CO. THAT HAVE A 0331;
PRIMARY MOS "
@ 22,0
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
RETURN
```


C. SELECT ATTRIBUTE ROUTINE

```

ROUTINE NAME: SLCTATRI.PRG
MODULE NAME: NONSTANDARD REPORT GENERATOR
VERSION: 4.2.0
AUTHOR: R.E. PRUIETT
DATE: 12FEB84
VARIABLES USED: COUNT, SLCT1, SLCT2, SLCT3
VARIABLES MODIFIED: NONE
VARIABLES CREATED: REPEAT, MATRIBUT, MATRI1, ATRI1,
MATRI2, ATRI2, MATRI3, ATRI3
VARIABLES RELEASED: NONE
FILES OPENED/CLOSED: NONE
TEMP FILES CREATED: NONE
USING SUBROUTINES: NCSTDRPT.FRG (4.0)
DESCRIPTION: THIS ROUTINE PROMPTS THE USER TO SELECT
ATTRIBUTES. AFTER THE FIRST AND SECOND SELECTION THE
USER IS ASKED IF HE WANTS TO SELECT ANOTHER ATTRIBUTE.
IF THE RESPONSE IS YES, THE SELECTION PROCEDURE IS
REPEATED. THIS PROCESS IS CONTINUED UNTIL THE USER
ANSWERS NC OR THREE ATTRIBUTES HAVE BEEN SELECTED. THE
ROUTINE WILL AUTOMATICALLY SORT AND LOCATE THE
APPROPRIATE DATABASE FILES AND DISPLAY THE NAME, SSN AND
ATTRIBUTE INFORMATION FOR EACH RECORD THAT SATISFIES THE
USER SPECIFIED CRITERIA.

```

```

STORE T TO REPEAT
DO WHILE REPEAT = T
STORE F TO REPEAT
ERASE
IF COUNT = 2 .OR. COUNT = 3
DO CASE

```

CASE SICT1 = "A"
@ 1,1 SAY "*"
CASE SICT1 = "B"
@ 2,1 SAY "*"
CASE SICT1 = "C"
@ 3,1 SAY "*"
CASE SICT1 = "D"
@ 4,1 SAY "*"
CASE SICT1 = "E"
@ 5,1 SAY "*"
CASE SICT1 = "F"
@ 6,1 SAY "*"
CASE SICT1 = "G"
@ 7,1 SAY "*"
CASE SICT1 = "H"
@ 8,1 SAY "*"
CASE SICT1 = "I"
@ 9,1 SAY "*"
CASE SICT1 = "J"
@ 10,1 SAY "*"
CASE SICT1 = "K"
@ 11,1 SAY "*"
CASE SICT1 = "L"
@ 12,1 SAY "*"
CASE SICT1 = "M"
@ 13,1 SAY "*"
CASE SICT1 = "N"
@ 14,1 SAY "*"
@ 15,1 SAY "*****"

```

ENDCASE
ENDIF
IF COUNT = 3
DO CASE
CASE SICT2 = "A"
  @ 1,1 SAY "*"
CASE SICT2 = "E"
  @ 2,1 SAY "*"

```



```

CASE SICT2 = "C"
@ 3,1 SAY "*"
CASE SICT2 = "D"
@ 4,1 SAY "*"
CASE SICT2 = "E"
@ 5,1 SAY "*"
CASE SICT2 = "F"
@ 6,1 SAY "*"
CASE SICT2 = "G"
@ 7,1 SAY "*"
CASE SICT2 = "H"
@ 8,1 SAY "*"
CASE SICT2 = "I"
@ 9,1 SAY "*"
CASE SICT2 = "J"
@ 10,1 SAY "*"
CASE SICT2 = "K"
@ 11,1 SAY "*"
CASE SICT2 = "L"
@ 12,1 SAY "*"
CASE SICT2 = "M"
@ 13,1 SAY "*"
CASE SICT2 = "N"
@ 14,1 SAY "*"
ENDCASE
ENDIF
@ 0,29 SAY "SELECT ATTRIBUTE"
@ 1,2 SAY "A...RANK"
@ 2,2 SAY "B...PRIMARY MOS"
@ 3,2 SAY "C...SECONDARY MOS"
@ 4,2 SAY "D...COMPANY"
@ 5,2 SAY "E...PLATCCN"
@ 6,2 SAY "F...JOIN DATE"
@ 7,2 SAY "G...EAS"
@ 8,2 SAY "H...BIRTH DATE"
@ 9,2 SAY "I...WEIGHT CONTROL"
@ 10,2 SAY "J...PFT CNE"
@ 11,2 SAY "K...PFT TWO"
@ 12,2 SAY "L...SWIM QUALIFICATION"
@ 13,2 SAY "M...RIFLE QUALIFICATION"
@ 14,2 SAY "N...PISTCI QUALIFICATION"
@ 22,0
ACCEPT "SELECT ATTRIBUTE ==>" TO MATRIBUT
IF CCUNT = 1
STORE MATRIBUT TO SICT1
DO CASE
CASE !(SLCT1) = "A"
STORE 'RANK' TO MATRI1
STORE 'RANK' TO ATRI1
CASE !(SICT1) = "E"
STORE 'PRIMARY MOS' TO MATRI1
STORE 'PRIMEMOS' TO ATRI1
CASE !(SICT1) = "C"
STORE 'SECONDARY MOS' TO MATRI1
STORE 'SECMOS' TO ATRI1
CASE !(SLCT1) = "D"
STORE 'COMPANY' TO MATRI1
STORE 'COMPANY' TO ATRI1
CASE !(SLCT1) = "F"
STORE 'PLATCON' TO MATRI1
STORE 'PLATOON' TO ATRI1
CASE !(SICT1) = "F"
STORE 'JOINDATE' TO MATRI1
STORE 'JOINDATE' TO ATRI1
CASE !(SICT1) = "G"
STORE 'EAS' TO MATRI1
STORE 'EAS' TO ATRI1
CASE !(SLCT1) = "H"
STORE 'BIRTH DATE' TO MATRI1

```



```

        STORE 'BRTHDATE' TO ATRI1
CASE !(SLCT1) = "I"
    STORE 'WEIGHT CCNTROL' TC MATRI1
    STORE 'WTCONT' TC ATRI1
CASE !(SLCT1) = "J"
    STORE 'PFT CNE' TO MATRI1
    STORE 'PFT1CLSS' TO ATRI1
CASE !(SLCT1) = "K"
    STORE 'PFT TWO' TO MATRI1
    STORE 'PFT2CLSS' TO ATRI1
CASE !(SLCT1) = "I"
    STORE 'SWIM QUALIFICATION' TO MATRI1
    STORE 'SWIMQUAL' TO ATRI1
CASE !(SLCT1) = "M"
    STORE 'RIFLE QUALIFICATION' TO MATRI1
    STORE 'RFLQUAL' TC ATRI1
CASE !(SLCT1) = "N"
    STORE 'PISTOL QUALIFICATION' TO MATRI1
    STORE 'PSTQUAL' TO ATRI1
OTHERWISE
    STORE T TO REPEAT
ENDCASE
ENDIF

```

```

IF CCNT = 2
    STORE MAIRIBUT TO SICT2
    DO CASE
        CASE !(SLCT2) = "A"
            STORE 'RANK' TO MATRI2
            STORE 'RANK' TO ATRI2
        CASE !(SLCT2) = "E"
            STORE 'PRIMARY MOS' TO MATRI2
            STORE 'PRIMEMOS' TO ATRI2
        CASE !(SLCT2) = "C"
            STORE 'SECONDARY MOS' TO MATRI2
            STORE 'SECMOS' TC ATRI2
        CASE !(SLCT2) = "D"
            STORE 'COMPANY' TO MATRI2
            STORE 'COMPANY' TC ATRI2
        CASE !(SLCT2) = "E"
            STORE 'PLATCON' TO MATRI2
            STORE 'PLATCON' TO ATRI2
        CASE !(SLCT2) = "F"
            STORE 'JOINDATE' TO MATRI2
            STORE 'JOINDATE' TO ATRI2
        CASE !(SLCT2) = "G"
            STORE 'EAS' TO MATRI2
            STORE 'EAS' TO ATRI2
        CASE !(SLCT2) = "F"
            STORE 'BIRTH DATE' TO MATRI2
            STORE 'BRTHDATE' TO ATRI2
        CASE !(SLCT2) = "I"
            STORE 'WEIGHT CCNTROL' TC MATRI2
            STORE 'WTCONT' TC ATRI2
        CASE !(SLCT2) = "J"
            STORE 'PFT ONE' TO MATRI2
            STORE 'PFT1CLSS' TO ATRI2
        CASE !(SLCT2) = "K"
            STORE 'PFT TWO' TO MATRI2
            STORE 'PFT2CLSS' TO ATRI2
        CASE !(SLCT2) = "I"
            STORE 'SWIM QUALIFICATION' TO MATRI2
            STORE 'SWIMQUAL' TO ATRI2
        CASE !(SLCT2) = "M"
            STORE 'RIFLE QUALIFICATION' TO MATRI2
            STORE 'RFLQUAL' TO ATRI2
        CASE !(SLCT2) = "N"
            STORE 'PISTOL QUALIFICATION' TO MATRI2
            STORE 'PSTQUAL' TO ATRI2
    
```



```

    OTHERWISE
      STORE T TO REPEAT
    ENDCASE
  ENDIF

IF COUNT = 3
  STORE MATRIBUT TO SICT3
  DO CASE
    CASE ! (SICT3) = "A"
      STORE 'RANK' TO MATRI3
      STORE 'RANK' TO ATRI3
    CASE ! (SICT3) = "E"
      STORE 'PRIMARY MOS' TO MATRI3
      STORE 'PRIMEMOS' TO ATRI3
    CASE ! (SICT3) = "C"
      STORE 'SECONDARY MOS' TO MATRI3
      STORE 'SECMOS' TC ATRI3
    CASE ! (SICT3) = "D"
      STORE 'COMPANY' TC MATRI3
      STORE 'COMPANY' TC ATRI3
    CASE ! (SICT3) = "E"
      STORE 'PLATCON' TC MATRI3
      STORE 'PLATOON' TO ATRI3
    CASE ! (SICT3) = "F"
      STORE 'JOINDATE' TO MATRI3
      STORE 'JOINDATE' TO ATRI3
    CASE ! (SICT3) = "G"
      STORE 'EAS' TO MATRI3
      STORE 'EAS' TO ATRI3
    CASE ! (SICT3) = "E"
      STORE 'BIRTH DATE' TO MATRI3
      STORE 'BRTHDATE' TO ATRI3
    CASE ! (SICT3) = "I"
      STORE 'WEIGHT CONTROL' TC MATRI3
      STORE 'WTCONT' TC ATRI3
    CASE ! (SICT3) = "J"
      STORE 'PFT ONE' TO MATRI3
      STORE 'PFT1CLSS' TO ATRI3
    CASE ! (SICT3) = "K"
      STORE 'PFT TWO' TO MATRI3
      STORE 'PFT2CLSS' TO ATRI3
    CASE ! (SICT3) = "I"
      STORE 'SWIM QUALIFICATION' TO MATRI3
      STORE 'SWIMQUAL' TO ATRI3
    CASE ! (SICT3) = "M"
      STORE 'RIFLE QUALIFICATION' TO MATRI3
      STORE 'RFLQUAL' TO ATRI3
    CASE ! (SICT3) = "N"
      STORE 'PISTOL QUALIFICATION' TO MATRI3
      STORE 'PSTQUAL' TO ATRI3
    OTHERWISE
      STORE T TO REPEAT
  ENDCASE
ENDIF
DO SICTDESC
ENDDC
RETURN

```


D. SELECT DESCRIPTOR ROUTINE

```

* ROUTINE NAME: SLCTDESC.PRG
* MODULE NAME: NONSTANDARD REPORT GENERATOR
* VERSION: 4.2.1.0
* AUTHCR: R.E. PRUIETT
* DATE: 14FEB84
* VARIABLES USED: COUNT,
* VARIABLES MODIFIED: DESC1, DESC2, DESC3
* VARIABLES CREATED: MRANK, MSEC MOS, MCOMPANY, MPLATOON,
* MJOINDTE, MEAS, MWTCNTL, MPFTONE, MPFTTWO, MSWMQUAL,
* MRIFOAL, MPISQAL, MRANK2, MPRIMOS2, MSEC MOS2, MCOMPNY2,
* MPLAT2, MJNDTE2, MEAS2, MBIRDTE2, MWTCNTL2, MPFTONE2,
* MPFTTWC2, MSWMQAL2, MRIFOAL2, MPISQAL2, MRANK3, MPRIMOS3,
* MSEC MOS3, MCOMPNY3, MPLAT3, MJNDTE3, MEAS3, MBIRDTE3,
* MWTCNTI3, MPFTCNE3, MPFTTWC3, MSWMQAL3, MRIFOAL3,
* MPISQAL3,
* VARIABLES RELEASED: NONE
* FILES CPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: SLCTATRI.PRG (4.2.0)
* DESCRIPTION: THIS ROUTINE PROMPTS THE USER AFTER EACH
* REQUEST FOR A NEW ATTRIBUTE TO SELECT THE DESCRIPTION FOR
* THE CORRESPONDING ATTRIBUTE. FOR EXAMPLE, IF THE USER
* HAS SELECTED THE ATTRIBUTE COMPANY, THIS ROUTINE WILL ASK
* WHICH COMPANY AND PROVIDES A RANGE/EXAMPLE OF APPROPRIATE
* DESCRIPTIONS (E.G., A CO, B CO, ETC).
ERASE
IF COUNT = 1
DO CASE
CASE !(SLCT1) = "A"
STORE ' ' TO MRANK
@ 8,0 SAY "TYPE IN DESIRED RANK DESCRIPTION";
"(E1,E5,02,ETC.)"
@ 8,48 GET MRANK PICTURE 'XX'
READ
STORE MRANK TO DESC1
CASE !(SLCT1) = "B"
STORE ' ' TC MPRIMOS
@ 8,0 SAY "TYPE IN DESIRED PRIMARY MOS DESCRIPTION";
"(0311,0331,ETC.)"
@ 8,58 GET MPRIMOS PICTURE 'XXXX'
READ
STORE MPRIMOS TC DESC1
CASE !(SLCT1) = "C"
STORE ' ' TC MSEC MOS
@ 8,0 SAY "TYPE IN DESIRED SECONDARY MOS DESCRIPTION";
"(0311,0331,ETC.)"
@ 8,60 GET MSEC MOS PICTURE 'XXXX'
READ
STORE MSEC MOS TC DESC1
CASE !(SLCT1) = "D"
STORE ' ' TC MCOMPANY
@ 8,0 SAY "TYPE IN DESIRED COMPANY DESCRIPTION (A";
"CO, WENS CO, H&S CO)"
@ 8,58 GET MCOMPANY PICTURE 'XXXXXX'
READ
STORE MCOMPANY TO DESC1
CASE !(SLCT1) = "E"
STORE ' ' TO MPLATOON
@ 8,0 SAY "TYPE IN DESIRED PLATOON DESCRIPTION (1ST";
"PLAT, 3D PLAT, ETC)"
@ 8,60 GET MPLATCON PICTURE 'XXXXXXXXXX'
READ
STORE MPLATOON TO DESC1
CASE !(SLCT1) = "F"
STORE ' ' TC MJOINDTE
@ 8,0 SAY "TYPE IN DESIRED JOIN DATE DESCRIPTION";

```



```

" (220183,0183,83,ETC.) "
@ 8,64 GET MJOINTE PICTURE 'XXXXXX'
READ
STORE MJOINTE TO DESC 1
CASE ! (SICT1) = "G"
STORE ' ' TO MEAS
@ 8,0 SAY "TYPE IN DESIRED EAS DESCRIPTION";
" (220183,0183,83,ETC.) "
@ 8,55 GET MEAS PICTURE 'XXXXXX'
READ
STORE MEAS TO DESC 1
CASE ! (SICT1) = "H"
STORE ' ' TO MBIRDATE
@ 8,0 SAY "TYPE IN DESIRED BIRTH DATE DESCRIPTION";
" (220183,0183,83,ETC.) "
@ 8,60 GET MBIRDATE PICTURE 'XXXXXX'
READ
STORE MBIRDATE TO DESC 1
CASE ! (SICT1) = "I"
STORE ' ' TO MWTCONTL
@ 8,0 SAY "TYPE IN DESIRED WEIGHT CONTROL DESCRIPTION";
"(YES CR NO)"
@ 8,55 GET MWTCONTL PICTURE 'XXX'
READ
STORE MWTCONTL TO DESC 1
CASE ! (SICT1) = "J"
STORE ' ' TO MPFTONE
@ 8,0 SAY "TYPE IN DESIRED PFT ONE DESCRIPTION";
"(S,1,2,3,F,*)"
@ 8,50 GET MPFTONE PICTURE 'X'
READ
STORE MPFTONE TO DESC 1
CASE ! (SICT1) = "K"
STORE ' ' TO MPFTTWO
@ 8,0 SAY "TYPE IN DESIRED PFT TWO DESCRIPTION";
"(S,1,2,3,F,*)"
@ 8,50 GET MPFTTWO PICTURE 'X'
READ
STORE MPFTTWO TO DESC 1
CASE ! (SICT1) = "L"
STORE ' ' TO MSWMQUAL
@ 8,0 SAY "TYPE IN DESIRED SWIMMING QUALIFICATION";
"DESCRIPTION"
@ 9,0 SAY "(S1,S2,S3,UN,*)"
@ 9,17 GET MSWMQUAL PICTURE 'XX'
READ
STORE MSWMQUAL TO DESC 1
CASE ! (SICT1) = "M"
STORE ' ' TO MRIFQUAL
@ 8,0 SAY "TYPE IN DESIRED RIFLE QUALIFICATION";
"DESCRIPTION"
@ 9,0 SAY "(EX,SS,MM,UN,*)"
@ 9,18 GET MRIFQUAL PICTURE 'XX'
READ
STORE MRIFQUAL TO DESC 1
CASE ! (SICT1) = "N"
STORE ' ' TO MPISQUAL
@ 8,0 SAY "TYPE IN DESIRED PISTOL QUALIFICATION";
"DESCRIPTION"
@ 9,0 SAY "(EX,SS,MM,UN,*)"
@ 9,18 GET MPISQUAL PICTURE 'XX'
READ
STORE MPISQUAL TO DESC 1
END CASE
ENDIF
IF COUNT = 2
DO CASE
CASE ! (SICT2) = "A"
STORE ' ' TO MRANK2

```



```

@ 8,0 SAY "TYPE IN DESIRED RANK DESCRIPTION;
(E1,E5,02,ETC.)"
@ 8,48 GET MRANK2 PICTURE 'XX'
REAL
STORE MRANK2 TO DESC2
CASE ! (SLCT2) = "E"
STORE ' ' TC MPRIMCS2
@ 8,0 SAY "TYPE IN DESIRED PRIMARY MOS DESCRIPTION;
(0311,C331,ETC.)"
@ 8,58 GET MPRIMCS2 PICTURE 'XXXX'
REAL
STORE MPRIMCS2 TC DESC2
CASE ! (SLCT2) = "C"
STORE ' ' TC MSEC MCS2
@ 8,0 SAY "TYPE IN DESIRED SECONDARY MOS DESCRIPTION;
(0311,0331,ETC.)"
@ 8,60 GET MSEC MCS2 PICTURE 'XXXX'
REAL
STORE MSEC MOS2 TC DESC2
CASE ! (SLCT2) = "D"
STORE ' ' TC MCOMP NY2
@ 8,0 SAY "TYPE IN DESIRED COMPANY DESCRIPTION (A;
CO, WPNS CO, H&S CC)"
@ 8,58 GET MCOMP NY2 PICTURE 'XXXXXX'
REAL
STORE MCOMP NY2 TC DESC2
CASE ! (SLCT2) = "E"
STORE ' ' TO MPLAT2
@ 8,0 SAY "TYPE IN DESIRED PLATOON DESCRIPTION (1ST;
PLAT, 3D PLAT, ETC.)"
@ 8,60 GET MPLAT2 PICTURE 'XXXXXXXXXX'
REAL
STORE MPLAT2 TO DESC2
CASE ! (SLCT2) = "F"
STORE ' ' TC MJNDTE2
@ 8,0 SAY "TYPE IN DESIRED JOIN DATE DESCRIPTION;
(220183,0183,83,ETC.)"
@ 8,64 GET MJNDTE2 PICTURE 'XXXXXX'
REAL
STORE MJNDTE2 TC DESC2
CASE ! (SLCT2) = "G"
STORE ' ' TO MEAS2
@ 8,0 SAY "TYPE IN DESIRED EAS DESCRIPTION;
(220183,0183,83,ETC.)"
@ 8,55 GET MEAS2 PICTURE 'XXXXXX'
REAL
STORE MEAS2 TO DESC2
CASE ! (SLCT2) = "H"
STORE ' ' TC MBIRDTE2
@ 8,0 SAY "TYPE IN DESIRED BIRTH DATE DESCRIPTION;
(220183,0183,83,ETC.)"
@ 8,60 GET MBIRDTE2 PICTURE 'XXXXXX'
REAL
STORE MBIRDTE2 TC DESC2
CASE ! (SLCT2) = "I"
STORE ' ' TO MWTCNTL2
@ 8,0 SAY "TYPE IN DESIRED WEIGHT CONTROL DESCRIPTION;
(YES CR NO)"
@ 8,55 GET MWTCNTL2 PICTURE 'XXX'
REAL
STORE MWTCNTL2 TC DESC2
CASE ! (SLCT2) = "J"
STORE ' ' TO MPFTCNE2
@ 8,0 SAY "TYPE IN DESIRED PFT ONE DESCRIPTION;
(S,1,2,3,F,*)"
@ 8,50 GET MPFTCNE2 PICTURE 'X'
REAL
STORE MPFTONE2 TC DESC2
CASE ! (SLCT2) = "K"

```



```

STORE ' ' TC MPFTTWO2
@ 8,0 SAY "TYPE IN DESIRED PFT TWO DESCRIPTION;
(S1,S2,S3,F,*)"
@ 8,50 GET MPFTTWO2 PICTURE 'X'
REAL
STORE MPFTTWO2 TC DESC2
CASE ! (SICT2) = "I"
STORE ' ' TO MSWMQAL2
@ 8,0 SAY "TYPE IN DESIRED SWIMMING QUALIFICATION;
DESCRIPTION"
@ 9,0 SAY " (S1,S2,S3,UN,**) "
@ 9,17 GET MSWMQAL2 PICTURE 'XX'
READ
STORE MSWMQAL2 TC DESC2
CASE ! (SICT2) = "R"
STORE ' ' TO MRIFQAL2
@ 8,0 SAY "TYPE IN DESIRED RIFLE QUALIFICATION;
DESCRIPTION"
@ 9,0 SAY " (EX,SS,MM,UN,**) "
@ 9,18 GET MRIFQAL2 PICTURE 'XX'
READ
STORE MRIFQAL2 TC DESC2
CASE ! (SICT2) = "N"
STORE ' ' TO MPISQAL2
@ 8,0 SAY "TYPE IN DESIRED PISTOL QUALIFICATION;
DESCRIPTION"
@ 9,0 SAY " (EX,SS,MM,UN,**) "
@ 9,18 GET MPISQAL2 PICTURE 'XX'
READ
STORE MPISQAL2 TC DESC2
END CASE
ENDIF
IF CCUNT = 3
IO CASE
CASE ! (SICT3) = "A"
STORE ' ' TO MRANK3
@ 8,0 SAY "TYPE IN DESIRED RANK DESCRIPTION;
(E1,E5,02,ETC.)"
@ 8,48 GET MRANK3 PICTURE 'XX'
REAL
STORE MRANK3 TO DESC3
CASE ! (SICT3) = "E"
STORE ' ' TC MPRIMCS3
@ 8,0 SAY "TYPE IN DESIRED PRIMARY MOS DESCRIPTION;
(0311,0331,ETC.)"
@ 8,58 GET MPRIMCS3 PICTURE 'XXXX'
REAL
STORE MPRIMCS3 TC DESC3
CASE ! (SICT3) = "C"
STORE ' ' TC MSEC MCS3
@ 8,0 SAY "TYPE IN DESIRED SECONDARY MOS DESCRIPTION;
(0311,0331,ETC.)"
@ 8,60 GET MSEC MCS3 PICTURE 'XXXX'
REAL
STORE MSEC MCS3 TC DESC3
CASE ! (SICT3) = "D"
STORE ' ' TC MCOMP NY3
@ 8,0 SAY "TYPE IN DESIRED COMPANY DESCRIPTION (A;
CO, WPNS CO, H&S CO)"
@ 8,58 GET MCOMP NY3 PICTURE 'XXXXXX'
REAL
STORE MCOMP NY3 TC DESC3
CASE ! (SICT3) = "F"
STORE ' ' TO MFLAT3
@ 8,0 SAY "TYPE IN DESIRED PLATOON DESCRIPTION (1ST;
FLAT, 3D PLAT, ETC)"
@ 8,60 GET MFLAT3 PICTURE 'XXXXXXXXXX'
REAL
STORE MFLAT3 TO DESC3

```



```

CASE ! (SLCT3) = "F"
STORE ' ' TC MJNDTE3
@ 8,0 SAY "TYPE IN DESIRED JOIN DATE DESCRIPTION;
(220183,0183,83,ETC.)"
@ 8,64 GET MJNDTE3 PICTURE 'XXXXXX'
REAL
STORE MJNDTE3 TC DESC3
CASE ! (SLCT3) = "G"
STORE ' ' TO MEAS3
@ 8,0 SAY "TYPE IN DESIRED EAS DESCRIPTION;
(220183,0183,83,ETC.)"
@ 8,55 GET MEAS3 PICTURE 'XXXXXX'
REAL
STORE MEAS3 TO DESC3
CASE ! (SLCT3) = "H"
STORE ' ' TC MBIRDTE3
@ 8,0 SAY "TYPE IN DESIRED BIRTH DATE DESCRIPTION;
(220183,0183,83,ETC.)"
@ 8,60 GET MBIRDTE3 PICTURE 'XXXXXX'
REAL
STORE MBIRDTE3 TC DESC3
CASE ! (SLCT3) = "I"
STORE ' ' TO MWTCNTL3
@ 8,0 SAY "TYPE IN DESIRED WEIGHT CONTROL DESCRIPTION;
(YES CR NO)"
@ 8,55 GET MWTCNTL3 PICTURE 'XXX'
REAL
STORE MWTCNTL3 TC DESC3
CASE ! (SLCT3) = "J"
STORE ' ' TO MPFTCNE3
@ 8,0 SAY "TYPE IN DESIRED PFT ONE DESCRIPTION;
(S,1,2,3,F,*)"
@ 8,50 GET MPFTCNE3 PICTURE 'X'
REAL
STORE MPFTONE3 TC DESC3
CASE ! (SLCT3) = "K"
STORE ' ' TO MPFTTWO3
@ 8,0 SAY "TYPE IN DESIRED PFT TWO DESCRIPTION;
(S,1,2,3,F,*)"
@ 8,50 GET MPFTTWC3 PICTURE 'X'
REAL
STORE MPFTTWO3 TC DESC3
CASE ! (SLCT3) = "L"
STORE ' ' TO MSWMQAL3
@ 8,0 SAY "TYPE IN DESIRED SWIMMING QUALIFICATION;
DESCRIPTION"
@ 9,0 SAY "(S1,S2,S3,UN,*)"
@ 9,17 GET MSWMQAL3 PICTURE 'XX'
REAL
STORE MSWMQAL3 TC DESC3
CASE ! (SLCT3) = "M"
STORE ' ' TO MRIFQAL3
@ 8,0 SAY "TYPE IN DESIRED RIFLE QUALIFICATION;
DESCRIPTION"
@ 9,0 SAY "(EX,SS,MM,UN,*)"
@ 9,18 GET MRIFQAL3 PICTURE 'XX'
REAL
STORE MRIFQAL3 TO DESC3
CASE ! (SLCT3) = "N"
STORE ' ' TO MPISQAL3
@ 8,0 SAY "TYPE IN DESIRED PISTOL QUALIFICATION;
DESCRIPTION"
@ 9,0 SAY "(EX,SS,MM,UN,*)"
@ 9,18 GET MPISQAL3 PICTURE 'XX'
REAL
STORE MPISQAL3 TC DESC3
END CASE
ENDIF
RETURN

```


APPENDIX G

STANDARD COMMAND GENERATOR MODULE LISTINGS

A. STANDARD REPORT ROUTINE

```
* ROUTINE NAME: STDREP.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.0
* AUTHCR: R.E. PRIETT
* DATE: 6NOV83
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: AASTDOPT, AADONE, PROCEED
* VARIABLES RELEASED: AASTDOPT, AADONE, ALL EXCEPT AA?????
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: COMMAND DETERMINER
* DESCRIPTION: GENERATES THE PRIMARY MENU FOR STANDARD
* REPORT SELECTION AND STORES THE USER'S RESPONSE IN
* AASTDOPT. CONTROL IS THEN PASSED TO THE APPROPRIATE
* SUBROUTINE TO OBTAIN THE NECESSARY DATA FROM THE
* DATABASE. THIS ROUTINE WILL CONTINUE UNTIL THE USER
* RESPONDS WITH A "Q".
```

```
STORE F TO AADONE
DO WHILE .NOT. AADONE
ERASE
```

```
* PRIMARY MENU FOR STANDARD REPORT SELECTION
@ 2,5 SAY " STANDARD REPORT SELECTION "
@ 3,5 SAY "SELECT A REPORT BY ENTERING THE APPROPRIATE;
LETTER "
@ 5,5 SAY "A...UNIT ROSTER "
@ 6,5 SAY "E...TRAINING STATUS REPORT "
@ 7,5 SAY "C...TRAINING ROSTER REPORT "
@ 8,5 SAY "D...INDIVIDUAL TRAINING RECORD "
@ 9,5 SAY "E...MARKSMANSHIP STATUS REPORT "
@ 10,5 SAY "F...MARKSMANSHIP ROSTER REPORT "
@ 11,5 SAY "G...MOS STATUS REPORT "
@ 12,5 SAY "H...MCS ROSTER REPORT "
@ 13,5 SAY "I...PERSONAL DATA REPORT "
@ 14,5 SAY "J...EST STATUS REPORT "
@ 15,5 SAY "K...EST ROSTER REPORT "
@ 16,5 SAY "L...SWIM QUALIFICATION REPORT "
@ 17,5 SAY "M...HELP "
@ 19,5 SAY "C...QUIT "
@ 20,5 SAY "
@ 21,5 SAY "
ACCEPT "SELECT OPTION =====> " TO AASTDOPT
```

```
DO CASE
CASE !(AASTDOPT) = 'A'
DC SLCTUNIT
DC UNITROST
ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE !(AASTDOPT) = 'E'
DC SLCTUNIT
DO TRNGSTAT
CASE !(AASTDOPT) = 'C'
DO SLCTUNIT
DO TRNGRCST
@ 22,0
```



```

ACCEPT "PRESS 'ENTER' KEY TC CONTINUE" TO PROCEED
CASE !(AASIDOPT) = 'C'
  ERASE
  DC VIEWITR
CASE !(AASIDOPT) = 'E'
  DO SLCTUNIT
  DC B:MKSSTAT
CASE !(AASIDOPT) = 'F'
  DO SLCTUNIT
  DO MKSROST
CASE !(AASIDOPT) = 'G'
  DO SLCTUNIT
  DO MOSSTAT
  @ 22,0
ACCEPT "PRESS 'ENTER' KEY TC CONTINUE" TO PROCEED
CASE !(AASIDOPT) = 'H'
  DO SLCTUNIT
  DO MOSROST
  @ 22,0
ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
CASE !(AASIDOPT) = 'I'
  DO SLCTUNIT
  DC PEFSDATA
  @ 22,0
ACCEPT "PRESS 'ENTER' KEY TC CONTINUE" TO PROCEED
CASE !(AASIDOPT) = 'J'
  DO SLCTUNIT
  DC ESTISTAT
CASE !(AASIDOPT) = 'K'
  DO SLCTUNIT
  DC ESTROST
  @ 22,0
ACCEPT "PRESS 'ENTER' KEY TC CONTINUE" TO PROCEED
CASE !(AASIDOPT) = 'L'
  DO SLCTUNIT
  DC SWIMQUAL
CASE !(AASIDOPT) = 'M'
  DO A:FELF
ENDCASE

IF !(AASIDCPT) = 'Q'
  STORE T TO AADCNE
ENDIF

ENDDC
RELEASE AADCNE, AASIDCPT
RELEASE ALL EXCEPT AA????
USE
RETURN

```


B. SELECT UNIT ROUTINE

```
* ROUTINE NAME: SLCTUNIT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.1.0
* AUTHOR: R.E. PRUIETT
* DATE: 6NOV83
* VARIABLES USED: ONICO, TWOCO, THREECO, FOURCO, ONEPLT,
* TWCPIT, THREEPIT, FCURPLT
* VARIABLES MODIFIED: NCNE
* VARIABLES CREATED: MUNIT, MPLAT
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING ROUTINES: STDRPT.PRG (5.0)
* DESCRIPTION: THIS ROUTINE PROMPTS THE USER TO SELECT
* THE SIZE AND NAME OF THE UNIT (I.E., BATTALION, COMPANY,
* PLATOON) FOR THE REPORT GENERATION. THE USER RESPONDS
* WITH A LETTER THAT CORRESPONDS TO A USER INSTALLED CC OR
* PLT DESIGNATION. THIS ROUTINE WILL EXECUTE ONCE EACH
* TIME CALLED.
```

```
STORE F TO SLCTDCNE
RESTORE FROM A:UNITMEM ADDITIVE
```

```
DO WHILE .NOT. SICTDCNE
```

```
ERASE
```

```
@ 6,5 SAY "SELECT UNIT
```

```
"
```

```
@ 8,5 SAY "A..."
```

```
@ 8,9 SAY CNECO
```

```
@ 10,5 SAY "B..."
```

```
@ 10,9 SAY TWOCO
```

```
@ 12,5 SAY "C..."
```

```
@ 12,9 SAY THREECO
```

```
@ 14,5 SAY "D..."
```

```
@ 14,9 SAY FCURCC
```

```
@ 16,5 SAY "E...BATTALION
```

```
"
```

```
@ 17,5 SAY "
```

```
"
```

```
@ 18,5 SAY "
```

```
"
```

```
ACCEPT "SELECT OPTION =====> " TO MUNIT
```

```
IF !(MUNIT)='A'.OR.!(MUNIT)='B'.OR.!(MUNIT)='C'.OR.;
```

```
!(MUNIT)='D'.OR.!(MUNIT)='E'
```

```
STORE T TO SLCTDONE
```

```
ENDIF
```

```
ENDDO
```

```
STORE F TO SLCTDONE
```

```
DO WHILE .NOT. SICTDCNE .AND. !(MUNIT)<>'E'
```

```
IF !(MUNIT)='A'.OR.!(MUNIT)='B'.OR.!(MUNIT)='C';
```

```
CR.!(MUNIT)='E'
```

```
ERASE
```

```
@ 10,5 SAY "SELECT SUBUNIT SIZE"
```

```
@ 12,5 SAY "A..."
```

```
@ 12,9 SAY ONEPLT
```

```
@ 14,5 SAY "B..."
```

```
@ 14,9 SAY TWOPLT
```

```
@ 16,5 SAY "C..."
```

```
@ 16,9 SAY THREEPIT
```

```
@ 18,5 SAY "D..."
```

```
@ 18,9 SAY FCURPIT
```

```
@ 20,5 SAY "E...ALL OF UNIT
```

```
"
```

```
@ 21,5 SAY "
```

```
"
```

```
@ 22,5 SAY "
```

```
"
```

```
ACCEPT "SELECT OPTION =====> " TO MPLAT
```

```
IF !(MPLAT)='A'.OR.!(MPLAT)='B'.OR.!(MPLAT)='C'.OR.;
```

```
!(MPLAT)='D'.OR.!(MPLAT)='E'
```

```
STORE T TO SLCTDONE
```



```
ENDIF  
ENDDC  
IF !(MUNIT) = 'E'  
    STCRE 'Z' TO MPLAT  
ENDIF  
RELEASE SLCTDONE  
USE  
FETURN
```


C. UNIT ROSTER ROUTINE

```
* ROUTINE NAME: UNITFCST.PRG
* MODULE NAME: STANDARD REPCRT GENERATOR
* VERSION: 5.2.0
* AUTHCR: R. E. FRUIETT
* DATE: 6NOV83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO,
* FOURCO, ONEPLT, TWOPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: MUNIT, MPLAT
* FILES (OPENED/CLOSED): A: PERS INDEX A:ALPHPERS
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTION: THIS IS A CONTRCL ROUTINE WHICH CREATES THE
* PERSONNEL ROSTER FOR THE UNIT SPECIFIED BY THE USER'S
* RESPONSE TO THE SLCTUNIT MENU(5.1.0).
```

```
ERASE
USE A:PEFS INDEX A:ALPHPERS
DO CASE
CASE !(MUNIT) = 'E'
REPORT FORM B:FCSTRPT
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'E'
REPORT FORM B:FCSTRPT FOR COMPANY = ONECO
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'A'
REPORT FORM B:FCSTRPT FOR COMPANY = ONECO .AND.;
PLATCCN = ONEPLT
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'B'
REPORT FORM B:FCSTRPT FOR COMPANY = ONECO .AND.;
PLATCCN = TWOPLT
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'C'
REPORT FORM B:FCSTRPT FOR COMPANY = ONECO .AND.;
PLATCCN = THREEPLT
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'D'
REPORT FORM B:FCSTRPT FOR COMPANY = ONECO .AND.;
PLATCCN = FOURPLT
CASE !(MUNIT) = 'E'.AND. !(MPLAT) = 'E'
REPORT FORM B:FCSTRPT FOR COMPANY = TWOCO
CASE !(MUNIT) = 'E'.AND. !(MPLAT) = 'A'
REPORT FORM B:FCSTRPT FOR COMPANY = TWOCO .AND.;
PLATCCN = ONEPLT
CASE !(MUNIT) = 'E'.AND. !(MPLAT) = 'B'
REPORT FORM B:FCSTRPT FOR COMPANY = TWOCO .AND.;
PLATCCN = TWOPLT
CASE !(MUNIT) = 'E'.AND. !(MPLAT) = 'C'
REPORT FORM B:FCSTRPT FOR COMPANY = TWOCO .AND.;
PLATCCN = THREEPLT
CASE !(MUNIT) = 'E'.AND. !(MPLAT) = 'D'
REPORT FORM B:FCSTRPT FOR COMPANY = TWOCO .AND.;
PLATCCN = FOURPLT
CASE !(MUNIT) = 'C'.AND. !(MPLAT) = 'E'
REPORT FORM B:FCSTRPT FOR COMPANY = THREECO
CASE !(MUNIT) = 'C'.AND. !(MPLAT) = 'A'
REPORT FORM B:FCSTRPT FOR COMPANY = THREECO .AND.;
PLATCCN = ONEPLT
CASE !(MUNIT) = 'C'.AND. !(MPLAT) = 'B'
REPORT FORM B:FCSTRPT FOR COMPANY = THREECO .AND.;
PLATCCN = TWOPLT
CASE !(MUNIT) = 'C'.AND. !(MPLAT) = 'C'
REPORT FORM B:FCSTRPT FOR COMPANY = THREECO .AND.;
PLATCCN = THREEPLT
CASE !(MUNIT) = 'C'.AND. !(MPLAT) = 'D'
REPORT FORM B:FCSTRPT FOR COMPANY = THREECO .AND.;
PLATCCN = FOURPLT
CASE !(MUNIT) = 'D'.AND. !(MPLAT) = 'E'
REPORT FORM B:FCSTRPT FOR COMPANY = FOURCO
```



```

CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'A'
  REPORT FORM B:FCSTRPT FOR COMPANY = FOURCO .AND.;
  PLATCCN = CNEPIT
CASE !(MUNIT) = 'I' .AND. !(MPLAT) = 'B'
  REPORT FORM B:FCSTRPT FOR COMPANY = FOURCO .AND.;
  PLATCCN = TWOPIIT
CASE !(MUNIT) = 'L' .AND. !(MPLAT) = 'C'
  REPORT FORM B:FCSTRPT FOR COMPANY = FOURCO .AND.;
  PLATCCN = THREEFLT
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'D'
  REPORT FORM B:FOSTRPT FOR COMPANY = FOURCO .AND.;
  PLATCCN = FOURFIT

```

ENDCASE

USE

RELEASE MUNIT, MPLAT

RETURN

D. TRAINING STATISTICS ROUTINE

```
* ROUTINE NAME: TRNGSTAT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.3.0
* AUTHCR: R. E. PRUIETT
* DATE: 22DEC83
* VARIABLES USED: MUNIT, MPLAT, ZZACOC, ZZTOTAL, ZZBCOC,
* ZZCCOC, ZZAHIS, ZZEHIS, ZZCHIS, ZZACOD, ZZBCOD, ZZCCOC,
* ZZAINT, ZZBINT, ZZCINT, ZZAAID, ZZBAID, ZZCAID, ZZDAID,
* ZZEAID, ZZFAID, ZZGAID, ZZHAID, ZZIAID, ZZJAID, ZZKAID,
* PROCEFF, ZZAUNI, ZZEUNI, ZZCUNI, ZZANBC, ZZBNBC, ZZCNBC,
* ZZDNBC, ZZFNBC, ZZFMKS, ZZAMKS, ZZBMKS, ZZCMKS, ZZDMKS,
* ZZEMKS, ZZFMKS, ZZATAC, ZZBTAC, ZZCTAC, ZZDTAC, ZZETAC,
* ZZPFT1, ZZPFT2, ZZSWIM, ZZRFL, ZZPST, ZZNAPST, ZZNARFL
* VARIABLES MODIFIED: NO GLCEAL VARIABLES MODIFIED
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: ZZ??????
* FILES (OPENED/CLOSED): EST, PERS, QUAL (ALL FILES CLOSED)
* TEMP FILES CREATED: TEMP
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTION: FOR THE SELECTED UNIT, THIS ROUTINE
* CALCULATES THE TOTAL NUMBER OF TRAINING ELEMENTS
* COMPLETED FOR EACH TRAINING CATEGORY. FOR EXAMPLE, THE
* TRAINING CATEGORY "HISTORY" CONSIST OF THREE TRAINING
* ELEMENTS (A, B, C). A PERCENTAGE COMPLETED FIGURE IS
* CALCULATED FOR EACH TRAINING ELEMENT BY DIVIDING THE
* TOTAL TRAINING ELEMENTS COMPLETED (EX. ZZAHIS, ZZEHIS,
* ZZCHIS) BY THE TOTAL NUMBER OF PERSONNEL IN THE
* UNIT(ZZTOTAL). BOTH THE ABSOLUTE TOTALS FOR EACH ELEMENT
* AND THE PERCENTAGE COMPLETED FIGURES ARE SHOWN.
```

```
ERASE
2 12,35 SAY "WORKING"
DO CASE
CASE !(MUNIT) = 'E'
USE B:EST
DO CNITRNG
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.COMPANY = ONECO .AND. P.SSN=S.SSN)
FIELDS COC,HIS,COD,INT,AID,UNI,;
PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNITRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.COMPANY = ONECO .AND. P.PLATOON =;
CNEPLT .AND. P.SSN=S.SSN) FIELDS;
COC,HIS,COD,INT,AID,UNI,PFT1CLSS,;
PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNITRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
```



```

SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =ONECO .AND. P.PLATOON =;
TWCPLT .AND. P.SSN=S.SSN) FIELDS;
COC,HIS,COD,INT,AID,UNI,PFT1CLSS;;
PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =ONECO .AND. P.PLATOON =;
THREEPLT .AND. P.SSN=S.SSN) FIELDS;
COC,HIS,COD,INT,AID,UNI,PFT1CLSS;;
PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =ONECO .AND. P.PLATOON =;
FOURPLT .AND. P.SSN=S.SSN) FIELDS;
COC,HIS,COD,INT,AID,UNI,PFT1CLSS;;
PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =TWOCO .AND.;
P.SSN=S.SSN) FIELDS COC,HIS,COD,INT;;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =TWOCO .AND. P.PLATOON =;
ONEPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD;;
INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =TWOCO .AND. P.PLATOON =;
TWCPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD;;
INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG

```



```

DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECNDARY
  USE B:EST
  SELECT PRIMARY
  JOIN TO TEMP FCR (P.COMPANY = TWOCO .AND. P.PLATOON =;
  THREEPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
  INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
  USE TEMP
  DO CNITRNG
  DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECNDARY
  USE B:EST
  SELECT PRIMARY
  JOIN TO TEMP FCR (P.COMPANY = TWOCO .AND. P.PLATOON =;
  FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
  INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
  USE TEMP
  DO CNITRNG
  DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECNDARY
  USE B:EST
  SELECT PRIMARY
  JOIN TO TEMP FCR (P.COMPANY = THREECO .AND. ;
  P.SSN=S.SSN) FIELDS CCC,HIS,COD,INT,;
  AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
  USE TEMP
  DO CNITRNG
  DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECNDARY
  USE B:EST
  SELECT PRIMARY
  JOIN TO TEMP FCR (P.COMPANY = THREECO .AND. P.PLATCON;
  = CNEFLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,;
  COL,INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
  USE TEMP
  DO CNITRNG
  DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECNDARY
  USE B:EST
  SELECT PRIMARY
  JOIN TO TEMP FCR (P.COMPANY = THREECO .AND. P.PLATCON;
  = TWCELT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
  INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
  USE TEMP
  DO CNITRNG
  DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECNDARY
  USE B:EST
  SELECT PRIMARY
  JOIN TO TEMP FCR (P.COMPANY = THREECO .AND. P.PLATCON;
  = THREEPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,;

```



```

INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =THREECO .AND. P.PIATCON;
= FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =FOURCO .AND.;
P.SSN=S.SSN) FIELDS CCC,HIS,COD,INT,;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =FOURCO .AND. P.PLATCCN;
= CNEPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =FOURCO .AND. P.PLATCCN;
= TWOPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,INT,;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =FOURCO .AND. P.PLATCCN;
= THREEPLT .AND. P.SSN=S.SSN) FIELDS;
COC,HIS,COD,INT,AID,UNI,PFT1CLSS,;
PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY

```



```

USE B:EST
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =FOURCO .AND. P.PLATCCN;
= FOURPLT .AND. P.SSN=S.SSN) FIELDS;
CCC,HIS,COD,INT,AID,UNI,PFT1CLSS,;
PFT2CLSS,NBC,MFS,TAC
USE TEMP
DO CNTTRNG
DELETE FILE TEMP

```

ENDCASE

ERASE

```

0,25 SAY "TRAINING STATUS REPORT"
2,0 SAY "CODE CF CCNDUCT..."
2,25 SAY (ZZACOC/ZZTOTAL)*100 USING '999'
2,30 SAY "PERCENT CCMPLETED A..."
2,52 SAY ZZACOC USING '999'
3,25 SAY (ZZBCOC/ZZTOTAL)*100 USING '999'
3,30 SAY "PERCENT CCMPLETED B..."
3,52 SAY ZZBCOC USING '999'
4,25 SAY (ZZCCOC/ZZTOTAL)*100 USING '999'
4,30 SAY "PERCENT CCMPLETED C..."
4,52 SAY ZZCCOC USING '999'
6,0 SAY "MARINE CORPS HISTORY..."
6,25 SAY (ZZAHIS/ZZTOTAL)*100 USING '999'
6,30 SAY "PERCENT CCMPLETED A..."
6,52 SAY ZZAHIS USING '999'
7,25 SAY (ZZBHIS/ZZTOTAL)*100 USING '999'
7,30 SAY "PERCENT CCMPLETED B..."
7,52 SAY ZZBHIS USING '999'
8,25 SAY (ZZCHIS/ZZTOTAL)*100 USING '999'
8,30 SAY "PERCENT CCMPLETED C..."
8,52 SAY ZZCHIS USING '999'
10,0 SAY "CLOSE ORDER DRILL..."
10,25 SAY (ZZACOD/ZZTOTAL)*100 USING '999'
10,30 SAY "PERCENT CCMPLETED A..."
10,52 SAY ZZACOD USING '999'
11,25 SAY (ZZBCOD/ZZTOTAL)*100 USING '999'
11,30 SAY "PERCENT CCMPLETED B..."
11,52 SAY ZZBCOD USING '999'
12,25 SAY (ZZCCOD/ZZTOTAL)*100 USING '999'
12,30 SAY "PERCENT CCMPLETED C..."
12,52 SAY ZZCCOD USING '999'
14,0 SAY "INTERIOR GUARD..."
14,25 SAY (ZZAINT/ZZTOTAL)*100 USING '999'
14,30 SAY "PERCENT CCMPLETED A..."
14,52 SAY ZZAINT USING '999'
15,25 SAY (ZZBINT/ZZTOTAL)*100 USING '999'
15,30 SAY "PERCENT CCMPLETED B..."
15,52 SAY ZZBINT USING '999'
16,25 SAY (ZZCINT/ZZTOTAL)*100 USING '999'
16,30 SAY "PERCENT CCMPLETED C..."
16,52 SAY ZZCINT USING '999'
18,0 SAY "FIRST AID..."
18,20 SAY (ZZAAID/ZZTOTAL)*100 USING '999'
18,24 SAY "PERCENT CCMPLETED A..."
18,46 SAY ZZAAID USING '999'
18,50 SAY (ZZBAID/ZZTOTAL)*100 USING '999'
18,54 SAY "PERCENT CCMPLETED B..."
18,76 SAY ZZBAID USING '999'
19,20 SAY (ZZCAID/ZZTOTAL)*100 USING '999'
19,24 SAY "PERCENT CCMPLETED C..."
19,46 SAY ZZCAID USING '999'
19,50 SAY (ZZDAID/ZZTOTAL)*100 USING '999'
19,54 SAY "PERCENT CCMPLETED D..."
19,76 SAY ZZDAID USING '999'
20,20 SAY (ZZEAID/ZZTOTAL)*100 USING '999'
20,24 SAY "PERCENT CCMPLETED E..."
20,46 SAY ZZEAID USING '999'
20,50 SAY (ZZFAID/ZZTOTAL)*100 USING '999'

```



```

20,54 SAY "PERCENT COMPLETED F..."
20,76 SAY ZZFAID USING '999'
21,20 SAY (ZZGAID/ZZTOTAL)*100 USING '999'
21,24 SAY "PERCENT COMPLETED G..."
21,46 SAY ZZGAID USING '999'
21,50 SAY (ZZHAID/ZZTOTAL)*100 USING '999'
21,54 SAY "PERCENT COMPLETED H..."
21,76 SAY ZZHAID USING '999'
22,20 SAY (ZZIAID/ZZTOTAL)*100 USING '999'
22,24 SAY "PERCENT COMPLETED I..."
22,46 SAY ZZIAID USING '999'
22,50 SAY (ZZJAID/ZZTOTAL)*100 USING '999'
22,54 SAY "PERCENT COMPLETED J..."
22,76 SAY ZZJAID USING '999'
23,20 SAY (ZZKAID/ZZTOTAL)*100 USING '999'
23,24 SAY "PERCENT COMPLETED K..."
23,46 SAY ZZKAID USING '999'
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ERASE
0,25 SAY "TRAINING STATUS REPORT"
2,0 SAY "EQUIPMENT/UNIFORMS..."
2,25 SAY (ZZAUNI/ZZTOTAL)*100 USING '999'
2,30 SAY "PERCENT COMPLETED A..."
2,52 SAY ZZAUNI USING '999'
3,25 SAY (ZZBUNI/ZZTOTAL)*100 USING '999'
3,30 SAY "PERCENT COMPLETED E..."
3,52 SAY ZZBUNI USING '999'
4,25 SAY (ZZCUNI/ZZTOTAL)*100 USING '999'
4,30 SAY "PERCENT COMPLETED C..."
4,52 SAY ZZCUNI USING '999'
6,0 SAY "NBC..."
6,25 SAY (ZZANEC/ZZTOTAL)*100 USING '999'
6,30 SAY "PERCENT COMPLETED A..."
6,52 SAY ZZANEC USING '999'
7,25 SAY (ZZBNEC/ZZTOTAL)*100 USING '999'
7,30 SAY "PERCENT COMPLETED E..."
7,52 SAY ZZBNEC USING '999'
8,25 SAY (ZZCNBC/ZZTOTAL)*100 USING '999'
8,30 SAY "PERCENT COMPLETED C..."
8,52 SAY ZZCNBC USING '999'
9,25 SAY (ZZDNBC/ZZTOTAL)*100 USING '999'
9,30 SAY "PERCENT COMPLETED D..."
9,52 SAY ZZDNBC USING '999'
10,25 SAY (ZZENBC/ZZTOTAL)*100 USING '999'
10,30 SAY "PERCENT COMPLETED E..."
10,52 SAY ZZENEC USING '999'
11,25 SAY (ZZFNBC/ZZTOTAL)*100 USING '999'
11,30 SAY "PERCENT COMPLETED F..."
11,52 SAY ZZFNBC USING '999'
13,0 SAY "MARKSMANSHIP..."
13,25 SAY (ZZAMKS/ZZTOTAL)*100 USING '999'
13,30 SAY "PERCENT COMPLETED A..."
13,52 SAY ZZAMKS USING '999'
14,25 SAY (ZZBMKS/ZZTOTAL)*100 USING '999'
14,30 SAY "PERCENT COMPLETED B..."
14,52 SAY ZZBMKS USING '999'
15,25 SAY (ZZCMKS/ZZTOTAL)*100 USING '999'
15,30 SAY "PERCENT COMPLETED C..."
15,52 SAY ZZCMKS USING '999'
16,25 SAY (ZZDMKS/ZZTOTAL)*100 USING '999'
16,30 SAY "PERCENT COMPLETED D..."
16,52 SAY ZZDMKS USING '999'
17,25 SAY (ZZEMKS/ZZTOTAL)*100 USING '999'
17,30 SAY "PERCENT COMPLETED E..."
17,52 SAY ZZEMKS USING '999'
18,25 SAY (ZZFMKS/ZZTOTAL)*100 USING '999'
18,30 SAY "PERCENT COMPLETED F..."
18,52 SAY ZZFMKS USING '999'
20,0

```



```

ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ERASE
@ 0,25 SAY "TRAINING STATUS REPORT"
@ 2,0 SAY "INDIVID TAC MEASURES..."
@ 2,25 SAY (ZZATAC/ZZTOTAL)*100 USING '999'
@ 2,30 SAY "PERCENT COMPLETED A..."
@ 2,52 SAY ZZATAC USING '999'
@ 3,25 SAY (ZZBTAC/ZZTOTAL)*100 USING '999'
@ 3,30 SAY "PERCENT COMPLETED B..."
@ 3,52 SAY ZZBTAC USING '999'
@ 4,25 SAY (ZZCTAC/ZZTOTAL)*100 USING '999'
@ 4,30 SAY "PERCENT COMPLETED C..."
@ 4,52 SAY ZZCTAC USING '999'
@ 5,25 SAY (ZZDTAC/ZZTOTAL)*100 USING '999'
@ 5,30 SAY "PERCENT COMPLETED D..."
@ 5,52 SAY ZZDTAC USING '999'
@ 6,25 SAY (ZZETAC/ZZTOTAL)*100 USING '999'
@ 6,30 SAY "PERCENT COMPLETED E..."
@ 6,52 SAY ZZETAC USING '999'
@ 8,0 SAY "PFT1..."
@ 8,24 SAY ((ZZTOTAL-ZZPFT1)/ZZTOTAL)*100 USING '999'
@ 8,31 SAY "PERCENT COMPLETED PFT ONE..."
@ 8,62 SAY (ZZTOTAL-ZZPFT1) USING '999'
@ 10,0 SAY "PFT2..."
@ 10,24 SAY ((ZZTOTAL-ZZPFT2)/ZZTOTAL)*100 USING '999'
@ 10,31 SAY "PERCENT COMPLETED PFT TWO..."
@ 10,62 SAY (ZZTOTAL-ZZPFT2) USING '999'
RESTORE FROM ZZTEMP
DO CASE
CASE !(MUNIT) = 'E'
USE B:QUAL
DO CNTQUAL
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:QUAL
SELECT PRIMARY
JCIN TO TEMP FOR (P.COMPANY = ONECO .AND.;
P.SSN=S.SSN) FIELDS SSN,SWIMQUAL, RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JCIN TO TEMP FOR (P.COMPANY = ONECO .AND. P.PLATOON =;
CNEFLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JCIN TO TEMP FOR (P.COMPANY = ONECO .AND. P.PLATOON =;
TWCFLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'C'
SELECT PRIMARY

```



```

USE A:PERS
SELECT SECNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =ONECO .AND. P.PIATCON =;
THREEPLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =ONECO .AND. P.PIATCON =;
FOURPLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =TWOCO .AND.;
P.SSN=S.SSN) FIELDS SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =TWOCO .AND. P.PIATCON =;
CNEPLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =TWOCO .AND. P.PIATCON =;
TWCPLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =TWOCO .AND. P.PIATCON =;
THREEPLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP

```



```

CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FOR (P.COMPANY = TWOCO .AND. P.PLATCON =;
  FOURPLT .AND. P.SSN=S.SSN) FIELDS;
  SSN,SWIMQUAL,RFLQUAL,PSTQUAL
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FOR (P.COMPANY = THREECO .AND.;
  P.SSN=S.SSN) FIELDS SSN,SWIMQUAL,RFLQUAL,PSTQUAL
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FOR (P.CCOMPANY = THREECO .AND. P.PIATCON;
  = CNEFLT .AND. P.SSN=S.SSN) FIELDS;
  SSN,SWIMQUAL,RFLQUAL,PSTQUAL
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FOR (P.COMPANY = THREECO .AND. P.PIATCON;
  = TWCEFLT .AND. P.SSN=S.SSN) FIELDS;
  SSN,SWIMQUAL,RFLQUAL,PSTQUAL
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FOR (P.COMPANY = THREECO .AND. P.PIATCON;
  = THREEFLT .AND. P.SSN=S.SSN) FIELDS;
  SSN,SWIMQUAL,RFLQUAL,PSTQUAL
  USE TEMP
  DO CNTQUAL
  DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
  SELECT PRIMARY
  USE A:PERS
  SELECT SECONDARY
  USE E:QUAL
  SELECT PRIMARY
  JOIN TO TEMP FOR (P.COMPANY = THREECO .AND. P.PIATCON;
  = FOURFLT .AND. P.SSN=S.SSN) FIELDS;
  SSN,SWIMQUAL,RFLQUAL,PSTQUAL
  USE TEMP

```



```

DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =FOURCO .AND.;
P.SSN=S.SSN) FIELDS SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.CCMPANY =FOURCO .AND. P.PLATOON;
= CNEPLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFIQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =FOURCO .AND. P.PLATCCN;
= TWOPLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =FOURCO .AND. P.PLATCCN;
= THREEPLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFLQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECNDARY
USE E:QUAL
SELECT PRIMARY
JOIN TO TEMP FOR (P.COMPANY =FOURCO .AND. P.PLATCCN;
= FOURPLT .AND. P.SSN=S.SSN) FIELDS;
SSN,SWIMQUAL,RFIQUAL,PSTQUAL
USE TEMP
DO CNTQUAL
DELETE FILE TEMP
ENDCASE
@ 12,0 SAY "SWIMQUAL..."
@ 12,24 SAY ((ZZTOTAL-ZZSWIM)/ZZTOTAL)*100 USING '999'
@ 12,31 SAY "PERCENT SWIM QUALIFIED..."
@ 12,60 SAY (ZZTOTAL-ZZSWIM) USING '999'
@ 14,0 SAY "RIFLE QUAL..."
@ 14,24 SAY ((ZZTOTAL-ZZRFL-ZZNARFL)/ZZTOTAL)*100 USING;
'999'

```


@ 14,31 SAY "PERCENT FIFTE QUALIFIED..."
@ 14,60 SAY (ZZTOTAL-ZZRFL-ZZNARFL) USING '999'
@ 16,0 SAY "PISTOL QUAL..."
@ 16,24 SAY ((ZZTOTAL-ZZPST-ZZNAPST)/ZZTOTAL)*100 USING;
'999'
@ 16,31 SAY "PERCENT PISTOL QUALIFIED..."
@ 16,60 SAY (ZZTOTAL-ZZPST-ZZNAPST) USING '999'
@ 20,C
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
RELEASE ALL LIKE ZZ?????
USE
RETURN

E. TRAINING COUNT ROUTINE

```

* ROUTINE NAME: CNTTFNG.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.3.1.0
* AUTHOR: R.E. PRUIETT
* DATE: 6NOV83
* VARIABLES USED: ONECO, TWCCO, THREECO, FOURCO, ONEPLT,
* TWCPIT, THREEPLT, *FOURPLT
* VARIABLES MODIFIED: NCNE
* VARIABLES CREATED: ZZACOC, ZZBCOC, ZZCCOC, ZZAHIS,
* ZZBHIS, ZZCHIS, ZZACOD, ZZBCOD, ZZCCOD, ZZAINT, ZZBINT,
* ZZCAID, ZZAAID, ZZFAID, ZZDAID, ZZEAID, ZZFAID,
* ZZGAID, ZZHAID, ZZIAID, ZZJAID, ZZKAID, ZZAUNI, ZZBUNI,
* ZZCUNI, ZZPFT1, ZZPFT2, ZZANBC, ZZBNBC, ZZCNBC, ZZDNBC,
* ZZENBC, ZZFNBC, ZZAMKS, ZZEMKS, ZZCMKS, ZZDMKS, ZZEMKS,
* ZZFMKS, ZZATAC, ZZETAC, ZZCTAC, ZZDTAC, ZZETAC
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: ZZTEMP (SAVES ALL VARIABLES UPON
* ENTERING THIS ROUTINE)
* USING ROUTINES: TRNGSTAT.PRG
* DESCRIPTION: THIS ROUTINE COUNTS THE NUMBER OF ESSENTIAL
* SUBJECT TRAINING ELEMENTS THAT HAVE BEEN COMPLETED.

```

SAVE TO ZZTEMP

RELEASE ALL

COUNT TO ZZTOTAL

COUNT FOR 'A'\$(COC)	TC ZZACOC
COUNT FOR 'E'\$(COC)	TC ZZBCOC
COUNT FOR 'C'\$(COC)	TC ZZCCOC
COUNT FOR 'A'\$(HIS)	TC ZZAHIS
COUNT FOR 'E'\$(HIS)	TC ZZBHIS
COUNT FOR 'C'\$(HIS)	TC ZZCHIS
COUNT FOR 'A'\$(COD)	TC ZZACOD
COUNT FOR 'E'\$(COD)	TC ZZBCOD
COUNT FOR 'C'\$(COD)	TC ZZCCOD
COUNT FOR 'A'\$(INT)	TC ZZAINT
COUNT FOR 'E'\$(INT)	TC ZZBINT
COUNT FOR 'C'\$(INT)	TC ZZCINT
COUNT FOR 'A'\$(AID)	TC ZZAAID
COUNT FOR 'E'\$(AID)	TC ZZBAID
COUNT FOR 'C'\$(AID)	TC ZZCAID
COUNT FOR 'D'\$(AID)	TC ZZDAID
COUNT FOR 'E'\$(AID)	TC ZZEAID
COUNT FOR 'F'\$(AID)	TC ZZFAID
COUNT FOR 'G'\$(AID)	TC ZZGAID
COUNT FOR 'H'\$(AID)	TC ZZHAID
COUNT FOR 'I'\$(AID)	TC ZZIAID
COUNT FOR 'J'\$(AID)	TC ZZJAID
COUNT FOR 'K'\$(AID)	TC ZZKAID
COUNT FOR 'A'\$(UNI)	TC ZZAUNI
COUNT FOR 'E'\$(UNI)	TC ZZBUNI
COUNT FOR 'C'\$(UNI)	TC ZZCUNI
COUNT FOR 'A'\$(NBC)	TC ZZANBC
COUNT FOR 'E'\$(NBC)	TC ZZBNBC
COUNT FOR 'C'\$(NBC)	TC ZZCNBC
COUNT FOR 'D'\$(NBC)	TC ZZDNBC
COUNT FOR 'E'\$(NBC)	TC ZZENBC
COUNT FOR 'F'\$(NBC)	TC ZZFNBC
COUNT FOR 'A'\$(MKS)	TC ZZAMKS
COUNT FOR 'E'\$(MKS)	TC ZZBMKS
COUNT FOR 'C'\$(MKS)	TC ZZCMKS
COUNT FOR 'D'\$(MKS)	TC ZZDMKS
COUNT FOR 'E'\$(MKS)	TC ZZEMKS
COUNT FOR 'F'\$(MKS)	TC ZZFMKS
COUNT FOR 'A'\$(TAC)	TC ZZATAC
COUNT FOR 'E'\$(TAC)	TC ZZBTAC


```
COUNT FOR 'C'$ (TAC) TC ZZCTAC
COUNT FOR 'D'$ (TAC) TC ZZDTAC
COUNT FOR 'E'$ (TAC) TC ZZETAC
COUNT FOR '*'$ (PFT1CISS) TO ZZFFT1
COUNT FOR '*'$ (PFT2CISS) TO ZZPFT2
RETURN
```


F. QUALIFICATION COUNT ROUTINE

```
* ROUTINE NAME: CNTQUAL.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION:5.3.2.0
* AUTHOR: R.E. PRUIETT
* DATE: 08DEC83
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: ZZSWIM, ZZRFL, ZZNARFL, ZZPST, ZZNAPST
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: TRNGSTAT.PRG
* DESCRIPTION: THIS ROUTINE FIRST SAVES ALL MEMORY
* VARIABLES IN A MEMORY FILE (ZZTOTAL) AND THEN COUNTS THE
* NUMBER OF PERSONS THAT HAVE NOT COMPLETED SWIMQUAL,
* RFLQUAL, PSTQUAL AND THOSE PERSONS NOT REQUIRED TO
* QUALIFY WITH THE RIFLE OR PISTOL (NA).
*
```

```
COUNT TC ZZTOTAL
COUNT FOR '***'$(SWIMQUAL) TO ZZSWIM
COUNT FOR '***'$(RFLQUAL) TO ZZFL
COUNT FOR 'NA'$(RFLQUAL) TO ZZNARFL
COUNT FOR '***'$(PSTQUAL) TO ZZPST
COUNT FOR 'NA'$(PSTQUAL) TO ZZNAPST
RETURN
```


G. TRAINING ROSTER FCUTINE

```
* ROUTINE NAME: TRNGFCST.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.4.0
* AUTHCR: R. E. FRUIT
* DATE: 20DEC83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO,
* FOURCO, CNEPLT, TWOPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: COUNT, PROCEED
* VARIABLES RELEASED: NONE
* FILES (OPENED/CLOSED): A:PERS INDEX A:ALPHPERS (ALL FILES
* CLCSED)
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: SDRPT.PRG (5.0)
* DESCRIPTION: THIS FCUTINE PROVIDES THE "TRAINING ROSTER
* REPORT". IT STARTS AT THE TOP OF THE PERS.DBF AND LOCKS
* AT EACH RECORD INDIVIDUALLY. IF A RECORD CORRESPONDS TO
* THE UNIT SELECTED BY THE USER, THE ROUTINE TRNGRPT.PRG IS
* CALLED WHICH COLLECTS AND DISPLAYS THE TRAINING DATA THAT
* CORRESPONDS TO THE CURRENT PERS.DBF RECORD (E.G., IF THE
* USER WANTS TRAINING DATA ON EVERY MEMBER OF 1ST PLAT, A
* CO, TRNGFCST.PRG LOCATES EACH RECORD IN THE PERS.DBF, A
* WHICH BELONG TO 1ST PLAT, A CO. TRNGRPT.PRG THEN
* DISPLAYS THE TRAINING DATA FROM THE APPROPRIATE DATABASE
* FILES.). INFORMATION ON ONE INDIVIDUAL IS DISPLAYED IN
* SIX ROWS AND FOUR SETS OF DATA ARE DISPLAYED PER SCREEN.
* THE USER PRESS'S THE ENTER KEY TO SCROLL FORWARD TO EACH
* SUCCEEDING SET OF INDIVIDUAL DATA.
```

```
ERASE
@ 12,28 SAY "TRAINING ROSTER REPORT"
@ 22,0
ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
ERASE
SELECT PRIMARY
USE A:PERS INDEX A:ALPHPERS
GOTO TOP
STORE 0 TO COUNT
DO WHILE .NOT. EOF
DO CASE
CASE !(MUNIT) = 'E'
DO TRNGRPT
STORE COUNT + 1 TO COUNT
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'E'
IF COMPANY = ONECO
DO TRNGRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'A'
IF COMPANY = ONECO .AND. PLATOON = ONEPLT
DO TRNGRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'B'
IF COMPANY = ONECO .AND. PLATOON = TWOPLT
DO TRNGRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'C'
IF COMPANY = ONECO .AND. PLATOON = THREEPLT
DO TRNGRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'D'
IF COMPANY = ONECO .AND. PLATOON = FOURPLT
DO TRNGRPT
```



```

        STCRE CCUNT + 1 TO CCUNT
    ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
    IF COMPANY = TWCCO
        DO TRNGRPT
        STORE COUNT + 1 TO COUNT
    ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
    IF CCMPANY = TWCCO .AND. PLATOON = ONEPLT
        DO TRNGRPT
        STCRE CCUNT + 1 TO CCUNT
    ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
    IF COMPANY = TWOCO .AND. PLATOON = TWOPLT
        DO TRNGRPT
        STORE COUNT + 1 TO COUNT
    ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
    IF COMPANY = TWCCO .AND. PLATOON = THREEPLT
        DO TRNGRPT
        STORE CCUNT + 1 TO CCUNT
    ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
    IF COMPANY = TWCCO .AND. PLATOON = FOURPLT
        DO TRNGRPT
        STORE COUNT + 1 TO COUNT
    ENDIF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
    IF COMPANY = THREECO
        DO TRNGRPT
        STCRE CCUNT + 1 TO CCUNT
    ENDIF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
    IF CCMPANY = THREECO .AND. PLATOON = ONEPLT
        DO TRNGRPT
        STORE COUNT + 1 TO COUNT
    ENDIF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
    IF COMPANY = THREECO .AND. PLATOON = TWOPLT
        DO TRNGRPT
        STORE CCUNT + 1 TO CCUNT
    ENDIF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
    IF COMPANY = TEREEO .AND. PLATOON = THREEPLT
        DO TRNGRPT
        STORE COUNT + 1 TO COUNT
    ENDIF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
    IF CCMPANY = THREECO .AND. PLATOON = FOURPLT
        DO TRNGRPT
        STCRE CCUNT + 1 TO CCUNT
    ENDIF
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'E'
    IF COMPANY = FCURCO
        DO TRNGRPT
        STCRE CCUNT + 1 TO COUNT
    ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
    IF COMPANY = FCURCO .AND. PLATOON = ONEPLT
        DO TRNGRPT
        STORE CCUNT + 1 TO CCUNT
    ENDIF
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'B'
    IF COMPANY = FCURCO .AND. PLATOON = TWOPLT
        DO TRNGRPT
        STCRE COUNT + 1 TO COUNT
    ENDIF
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'C'
    IF CCMPANY = FCURCO .AND. PLATOON = THREEPLT

```



```

        DO TRNGRPT
        STORE CCUNT + 1 TO CCUNT
        ENDIF
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'D'
    IF COMPANY = FCURCO .AND. PLATOON = FOURPLT
        DO TRNGRPT
        STORE CCUNT + 1 TO CCUNT
        ENDIF
    ENDCASE
    IF CCUNT = 4
        @ 22,0
        ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
        STORE 0 TO COUNT
        ERASE
    ENDIF
    SELECT PFIMARY
    SKIP
    ENDDC
    RELEASE COUNT, PROCEED, MSSN
    USE
    RETURN

```


H. TRAINING ROSTER FORMAT ROUTINE

```
* ROUTINE NAME:
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.4.1.0
* AUTHOR: R.E. PRUIETT
* DATE: 15DEC83
* VARIABLES USED: COUNT, PLATNAME, UNITNAME, MPLAT, MUNIT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: FCW, MSSN
* VARIABLES RELEASED: NONE * FILES OPENED/CLOSED: E:QUAL
* INDEX E:QUALSSN, B:EST INDEX B:ESTSSN
* TEMP FILES CREATED:
* USING SUBROUTINES: TRNGROST.PRG
* DESCRIPTION: THIS ROUTINE PROVIDES THE FORMAT AND
* DISPLAYS THE "TRAINING ROSTER REPORT".
```

```
STORE (COUNT * 6) TO FOW
@ ROW,0 SAY NAME
@ ROW,30 SAY RANK
@ ROW,33 SAY COMPANY
STORE SSN TO MSSN
SELECT SECONDARY
USE B:QUAL INDEX B:QUALSSN
FIND 8MSSN
@ ROW,39 SAY "SWIM QUAL "
@ ROW,49 SAY "SWIMQUAL "
@ ROW,52 SAY "RIFLE QUAL "
@ ROW,63 SAY "RFLQUAL "
@ ROW,66 SAY "PISTOL QUAL "
@ ROW,78 SAY "PSTQUAL "
SELECT SECONDARY
USE E:EST INDEX B:ESTSSN
FIND 8MSSN
@ ROW + 1,0 SAY "CODE OF CONDUCT "
@ ROW + 1,16 SAY "COC "
@ ROW + 1,22 SAY "MARCOR HIS "
@ ROW + 1,35 SAY "HIS "
@ ROW + 1,43 SAY "MARKSMANSHIP "
@ ROW + 1,56 SAY "MKS "
@ ROW + 1,67 SAY "DRILL "
@ ROW + 1,74 SAY "COD "
@ ROW + 2,0 SAY "INTERIOR GUARD "
@ ROW + 2,16 SAY "INT "
@ ROW + 2,22 SAY "MARKSMANSHIP "
@ ROW + 2,35 SAY "MKS "
@ ROW + 2,43 SAY "TAC MEASURES "
@ ROW + 2,56 SAY "TAC "
@ ROW + 2,67 SAY "NBC "
@ ROW + 2,74 SAY "NBC "
@ ROW + 3,0 SAY "PFT1 "
@ ROW + 3,6 SAY "PFT1CISS "
@ ROW + 3,10 SAY "PFT2 "
@ ROW + 3,16 SAY "PFT2CLSS "
@ ROW + 3,22 SAY "EQUIP/UNIFORM "
@ ROW + 3,36 SAY "UNI "
@ ROW + 3,43 SAY "FIRST AID "
@ ROW + 3,54 SAY "AID "
RETURN
```


I. MARKSMANSHIP STATUS ROUTINE

```

* ROUTINE NAME: MKSSTAT.FRG
* MODULE NAME: STANDARD REPCRT GENERATOR
* VERSICN: 5.6.0
* AUTHCR: R.E. PRUIETT
* DATE: 28NCV83
* VARIABLES USED: UNITNAME, PLATNAME
* VARIABLES MODIFIED: NO GLOBAL VARIABLES MODIFIED
* VARIABLES CREATED: RFLX, RFLSS, RFLMM, RFLUN, PSTEX,
* PSTSS, PSTMM, PSTUN, RFLSHOT, PCTEX, PCTSS, PCTMM, PCTUN,
* PCTEXEST, PCTSSPST, PCTMMEST, PCTUNPST, PROCEED, ANALYZE6
* VARIABLES RELEASED: EST?????, PCT?????, RFLX, RFLSS,
* RFLMM, RFLUN, PROCEED, ANALYZE6
* FILES (OPENED/CLOSED): ALL CLCSED
* TEMP FILES CREATED: MRKSTEMP, MKSTEMP
* USING SUBROUTINES: STDRPT.FRG
* DESCRIPTION: THIS ROUTINE CREATES A TEMPORARY FILE WHICH
* CONTAINS THE CURRENT YEAR RIFLE AND PISTOL QUALIFICATION
* RESULTS FOR EACH MEMBER IN THE SELECTED COMPANY AND/OR
* PLATCCN. THE RESULTS ARE TOTALED BY QUALIFICATION
* CATEGORY FOR THE RIFLE AND PISTOL. PERCENTAGES ARE
* CALCULATED AND DISPLAYED IN A SUMMARY STATUS REPORT WHICH
* IS AUTOMATICALLY DISPLAYED ON THE SCREEN.
ERASE
DO CVFTUNIT
DO CASE
  CASE !(MUNIT)='E'
    USE B:QUAL
  CASE !(MUNIT)<>'F' .AND. !(MPLAT)='E'
    USE A:PERS
    CCPY TO MRKSTEMP ALL FIELD SSN FOR COMPANY=UNITNAME
    USE MRKSTEMP
    SELECT SECONDARY
    USE B:QUAL
    SELECT PRIMARY
    JCIN TO MKSTEMP FOR F.SSN = S.SSN FIELDS RFLQUAL,;
    PSTQUAL
    USE MKSTEMP
  CASE !(MUNIT)<>'F' .AND. !(MPLAT)<>'E'
    USE A:PERS
    CCPY TO MRKSTEMP ALL FIELD SSN FOR COMPANY=UNITNAME;
    .AND. PLATOON = PLATNAME
    USE MRKSTEMP
    SELECT SECONDARY
    USE B:QUAL
    SELECT PRIMARY
    JCIN TO MKSTEMP FOR F.SSN = S.SSN FIELDS RFLQUAL,;
    PSTQUAL
    USE MKSTEMP
ENDCASE
COUNT FOR RFLQUAL = 'EX' TO RFLX
COUNT FOR RFLQUAL = 'SS' TO RFLSS
COUNT FOR RFLQUAL = 'MM' TO RFLMM
COUNT FOR RFLQUAL = 'UN' TO RFLUN
COUNT FOR PSTQUAL = 'EX' TO PSTEX
COUNT FOR PSTQUAL = 'SS' TO PSTSS
COUNT FOR PSTQUAL = 'MM' TO PSTMM
COUNT FOR PSTQUAL = 'UN' TO PSTUN
ERASE
STORE (RFLX+RFLSS+RFLMM+RFLUN) TO RFLSHOT
IF RFLSHOT > 0
  STCFE (RFLX/RFLSHOT*100) TO PCTEX
  STCFE (RFLSS/RFLSHOT*100) TO PCTSS
  STCFE (RFLMM/RFLSHOT*100) TO PCTMM
  STCFE (RFLUN/RFLSHOT*100) TO PCTUN
ENDIF
STORE (PSTEX+PSTSS+PSTMM+PSTUN) TO PSTSHOT

```



```

IF PSTSHCT > 0
  STCFE (PSTEX/PSTSHCT*100) TO PCTEXPST
  STCFE (PSSI/PSTSHCT*100) TO PCTSSPST
  STCFE (PSTMM/PSTSHCT*100) TO PCTMMPST
  STCFE (PSTUN/PSTSHCT*100) TO PCTUNPST
ENDIF
@ 1,30 SAY "MARKSMANSHIP STATUS REPORT"
@ 3,1 SAY "RIFLE EXPERTS....."
@ 3,23 SAY RFLX
@ 3,39 SAY "PISTOL EXPERTS....."
@ 3,63 SAY PSTEX
@ 5,1 SAY "RIFLE SHAFESHOTTERS...."
@ 5,23 SAY RFLSS
@ 5,39 SAY "PISTOL SHARPSHOOTERS...."
@ 5,63 SAY PSTSS
@ 7,1 SAY "RIFLE MARKSMEN....."
@ 7,23 SAY RFLMM
@ 7,39 SAY "PISTOL MARKSMEN....."
@ 7,63 SAY PSTMM
@ 9,1 SAY "UNQUALIFIED....."
@ 9,23 SAY RFLUN
@ 9,39 SAY "UNQUALIFIED....."
@ 9,63 SAY PSTUN
@ 11,1 SAY "TOTAL....."
@ 11,23 SAY RFLSHOT
@ 11,39 SAY "TOTAL....."
@ 11,63 SAY PSTSHOT
@ 15,34 SAY "PERCENTAGES"
@ 16,34 SAY "-----"
IF RFLSHCT > 0
  @ 18,1 SAY "RIFLE EXPERTS....."
  @ 18,23 SAY PCTEX
  @ 19,1 SAY "RIFLE SHARPSHOTTERS....."
  @ 19,23 SAY PCTISS
  @ 20,1 SAY "RIFLE MARKSMEN....."
  @ 20,23 SAY PCTMM
  @ 21,1 SAY "UNQUALIFIED....."
  @ 21,23 SAY PCTUN
ELSE
  @ 18,1 SAY "NO UNIT MEMBERS HAVE FIRED THE "
  @ 20,1 SAY "RIFLE FOR QUALIFICATION THIS YEAR."
ENDIF
IF PSTSHCT > 0
  @ 18,39 SAY "PISTOL EXPERTS....."
  @ 18,63 SAY PCTEXPST
  @ 19,39 SAY "PISTOL SHARP SHOOTERS...."
  @ 19,63 SAY PCTSSPST
  @ 20,39 SAY "PISTOL MARKSMEN....."
  @ 20,63 SAY PCTMMPST
  @ 21,39 SAY "UNQUALIFIED....."
  @ 21,63 SAY PCTUNPST
ELSE
  @ 18,39 SAY "NO UNIT MEMBERS HAVE FIRED THE"
  @ 20,39 SAY "PISTOL FOR QUALIFICATION THIS YEAR"
ENDIF
@ 22,0
ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
RELEASE ALL LIKE PST????
RELEASE ALL LIKE PCT????
FRASE
ACCEPT "DO YOU WANT TO DO A COMPANY/BATTALION RANGE QUOTA;
ANALYSIS? (Y/N) " TO ANALYZE6
IF !(ANALYZE6) = 'Y'
  DC F:QTASTUDY
ENDIF
RELEASE RFLX,RFLSS,RFLMM,RFLUN,RFLS,PROCEED,ANALYZE6
USE
RETURN

```


J. CONVERT UNIT ROUTINE

```
* ROUTINE NAME: CVRTUNIT.PRG
* MODULE NAME: STANDARD REPCRT GENERATOR
* VERSION: 5.6.1.0
* AUTHOR: R.E. PRUIETT
* DATE: 28NOV83
* VARIABLES USED: MUNIT, MPIAT, ONECO, TWOCC, THREECO,
* FOURCO, ONEPLT, TWOPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: UNITNAME, PLATNAME
* VARIABLES RELEASED: NONE
* FILES (OPENED/CLOSED): NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: MKSSTAT.PRG (5.6.0) AND MOSSTAT.PRG
* (5.8.0)
* DESCRIPTION: THIS ROUTINE STORES THE NAME OF THE COMPANY
* AND PLATCON SELECTED BY THE USER DURING ROUTINE
* SLCTUNIT'. IT IS NECESSARY BECAUSE WHEN THE USER
* SELECTS A UNIT AND SUBUNIT IT IS DONE BY MENU SELECTION.
* ACCORDINGLY THE ONLY THING STORED INTO MEMORY IS THE
* CORRESPONDING LETTER FROM THE MENU PROVIDED BY THE
* SLCTUNIT.PRG (A, B, C, D, E). SOME SCREEN DISPLAYS
* REQUIRE THE UNIT NAME (E.G., "A CO" VICE "A") SO A
* CONVERSION IS NECESSARY.
```

```
DO CASE
  CASE !(MUNIT)='A'
    STCRE ONECO TO UNITNAME
  CASE !(MUNIT)='B'
    STCRE TWOCC TO UNITNAME
  CASE !(MUNIT)='C'
    STCRE THREECO TO UNITNAME
  CASE !(MUNIT)='D'
    STCRE FOURCO TO UNITNAME
ENDCASE
IF !(MUNIT) <> 'E'
DO CASE
  CASE !(MPLAT)='A'
    STCRE ONEPLT TO PLATNAME
  CASE !(MPLAT)='B'
    STCRE TWOPLT TO PLATNAME
  CASE !(MPLAT)='C'
    STCRE THREEPLT TO PLATNAME
  CASE !(MPLAT)='D'
    STCRE FOURPLT TO PLATNAME
ENDCASE
ENDIF
RETURN
```


K. QUOTA STUDY ROUTINE

```
* ROUTINE NAME: QTAstudy.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.6.2.0
* AUTHCR: R.E. PRUIETT
* DATE: 15DEC83
* VARIABLES USED: UNITNAME, PLATNAME, RFLSHOT, RFLEX,
* RFLSS, RFLMM, RFLUN, MUNIT, QFEB, QMAR,
* ... QDEC
* VARIABLES MODIFIED: NO GLOBAL VARIABLES MODIFIED
* VARIABLES CREATED: RFLNA, MCNTH, TOTALREC, MLEFT, MSUM,
* NEEDED, EXTRA, WHATIF
* VARIABLES RELEASED: MONTH, WHATIF, EXTRA, NEEDED, MSUM,
* MLEFT, TOTALREC, RFLNA, Q???
* FILES (OPENED/CLOSED): NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: MKSSTAT.PRG
* DESCRIPTION: THIS MODULE IS USED BY MKSSTAT.PRG AND
* DISPLAYS THE RANGE QUOTAS THAT REMAIN FOR THE CURRENT
* YEAR. THE CURRENT MONTH IS EXTRACTED FROM THE 'LOG ON
* DATE' AND THE RANGE QUOTAS FOR EACH UNIT IS STORED IN A
* MEMORY FILE DURING THE ANNUAL INITIALIZATION PROCEDURE.
* SUMMARY INFORMATION IS PROVIDED CONCERNING THE RELATION-
* SHIP BETWEEN REMAINING RANGE QUOTA REQUIREMENTS AND
* REMAINING RANGE QUOTA ALLOCATIONS. UPON CONCLUSION OF
* THE MODULE THE USER MAY RETURN TO THE STANDARD REPORT
* MENU OR TO A SCRATCH PAD TO REVISE QUOTA ASSIGNMENTS.
* REVISED QUOTA ASSIGNMENTS FROM THE SCRATCH PAD CAN EITHER
* BE SAVED OR IGNORED.
```

```
ERASE
DO CASE
CASE !(MUNIT) <> 'E' .AND. !(MPLAT) = 'E'
@ 1,15 SAY "RANGE QUOTA ANALYSIS FOR "
@ 1,40 SAY UNITNAME
CASE !(MUNIT) <> 'E' .AND. !(MPLAT) <> 'E'
@ 1,15 SAY "RANGE QUOTA ANALYSIS FOR "
@ 1,40 SAY UNITNAME
@ 1,46 SAY " "
@ 1,48 SAY PLATNAME
CASE !(MUNIT) = 'E'
@ 1,15 SAY "RANGE QUOTA ANALYSIS FOR BATTALION"
ENDCASE
COUNT FOR RFLQUAL = 'NA' TO RFLNA
STORE $(AADATE,3,2) TO MCNTH
GOTO EOTTCM
STORE # TO TOTALREC
@ 4,1 SAY "NUMBER OF MARINES FIRED FOR QUALIFICATION;
DURING"
@ 5,1 SAY "CURRENT;
YEAR....."
@ 5,50 SAY RFLSHOT
@ 6,1 SAY "NUMBER OF MARINES FIRED FOR QUALIFICATION;
DURING"
@ 7,1 SAY "CURRENT YEAR BUT DID NOT;
QUALIFY....."
@ 7,50 SAY RFLUN
@ 8,1 SAY "NUMBER OF MARINES NOT REQUIRED TO QUALIFY;
DURING"
@ 9,1 SAY "CURRENT;
YEAR....."
@ 9,50 SAY RFLNA
STORE (TOTALREC - RFLNA - RFLEX - RFLSS - RFLMM) TO MLEFT
@ 10,1 SAY "MARINES LEFT TO QUALIFY DURING CURRENT;
YEAR....."
@ 10,50 SAY MLEFT
DO CASE
```



```

CASE !(MUNIT) = 'A'
  RESTORE FROM A:ARNGQTAS ADDITIVE
CASE !(MUNIT) = 'E'
  RESTORE FROM A:ERNGQTAS ADDITIVE
CASE !(MUNIT) = 'C'
  RESTORE FROM A:CRNGQTAS ADDITIVE
CASE !(MUNIT) = 'D'
  RESTORE FROM A:DRNGQTAS ADDITIVE
CASE !(MUNIT) = 'E'
  RESTORE FROM A:ENRNGQTA ADDITIVE
ENDCASE
DO CASE
CASE MCNTH = '01'
  @ 12,4 SAY "FEB      MAR      APR      MAY      JUN      JUL      AUG;
  SEP      OCT      NOV      DEC"
  @ 14,4 SAY STR(CFEB,3)
  @ 14,10 SAY STR(CMAR,3)
  @ 14,16 SAY STR(CAPR,3)
  @ 14,22 SAY STR(QMAY,3)
  @ 14,28 SAY STR(QJUN,3)
  @ 14,34 SAY STR(QJUL,3)
  @ 14,40 SAY STR(QAUG,3)
  @ 14,46 SAY STR(QSEP,3)
  @ 14,52 SAY STR(QCCT,3)
  @ 14,58 SAY STR(QNCV,3)
  @ 14,64 SAY STR(QDEC,3)
  STCRE (QFEB+ QMAR+ QAPR+ QMAY+ QJUN+ QJUL+ QAUG+ QSEP+
  QCCT+ QNCV+ QDEC) TO MSUM
  @ 16,4 SAY "A TOTAL OF "
  @ 16,15 SAY STR(MSUM,3)
  @ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH = '02'
  @ 12,4 SAY "MAR      APR      MAY      JUN      JUL      AUG      SEP;
  OCT      NOV      DEC"
  @ 14,4 SAY STR(QMAR,3)
  @ 14,10 SAY STR(CAPR,3)
  @ 14,16 SAY STR(QMAY,3)
  @ 14,22 SAY STR(QJUN,3)
  @ 14,28 SAY STR(QJUL,3)
  @ 14,34 SAY STR(QAUG,3)
  @ 14,40 SAY STR(QSEP,3)
  @ 14,46 SAY STR(QCCT,3)
  @ 14,52 SAY STR(QNCV,3)
  @ 14,58 SAY STR(QDEC,3)
  STCRE (QMAR+ QAPR+ QMAY+ QJUN+ QJUL+ QAUG+ QSEP+ QOCT+
  QNCV+ QDEC) TO MSUM
  @ 16,4 SAY "A TOTAL OF "
  @ 16,15 SAY STR(MSUM,3)
  @ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH = '03'
  @ 12,4 SAY "APR      MAY      JUN      JUL      AUG      SEP      OCT;
  NOV      DEC"
  @ 14,4 SAY STR(QAPR,3)
  @ 14,10 SAY STR(QMAY,3)
  @ 14,16 SAY STR(QJUN,3)
  @ 14,22 SAY STR(QJUL,3)
  @ 14,28 SAY STR(QAUG,3)
  @ 14,34 SAY STR(QSEP,3)
  @ 14,40 SAY STR(QCCT,3)
  @ 14,46 SAY STR(QNCV,3)
  @ 14,52 SAY STR(QDEC,3)
  STCRE (QAPR+ QMAY+ QJUN+ QJUL+ QAUG+ QSEP+ QOCT+ QNOV+ QDEC) ;
  TO MSUM
  @ 16,4 SAY "A TOTAL OF "
  @ 16,15 SAY STR(MSUM,3)
  @ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH = '04'
  @ 12,4 SAY "MAY      JUN      JUL      AUG      SEP      OCT      NOV;
  DEC"

```



```

@ 14,4 SAY STR(QMAY,3)
@ 14,10 SAY STR(QJUN,3)
@ 14,16 SAY STR(QJUL,3)
@ 14,22 SAY STR(QAUG,3)
@ 14,28 SAY STR(QSEP,3)
@ 14,34 SAY STR(QCCT,3)
@ 14,40 SAY STR(QNOV,3)
@ 14,48 SAY STR(QDEC,3)
STCRE (QMAY+QJUN+QJUL+QAUG+QSEP+QOCT+QNOV+QDEC) TO MSUM
@ 16,4 SAY "A TOTAL OF "
@ 16,15 SAY STR(MSUM,3)
@ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH = '05'
@ 12,4 SAY "JUN JUL AUG SEP OCT NOV DEC"
@ 14,4 SAY STR(QJUN,3)
@ 14,10 SAY STR(QJUL,3)
@ 14,16 SAY STR(QAUG,3)
@ 14,22 SAY STR(QSEP,3)
@ 14,28 SAY STR(QCCT,3)
@ 14,34 SAY STR(QNOV,3)
@ 14,40 SAY STR(QDEC,3)
STCRE (QJUN+QJUL+QAUG+QSEP+QCCT+QNOV+QDEC) TO MSUM
@ 16,4 SAY "A TOTAL OF "
@ 16,15 SAY STR(MSUM,3)
@ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH = '06'
@ 12,4 SAY "JUL AUG SEP OCT NOV DEC"
@ 14,4 SAY STR(QJUL,3)
@ 14,10 SAY STR(QAUG,3)
@ 14,16 SAY STR(QSEP,3)
@ 14,22 SAY STR(QOCT,3)
@ 14,28 SAY STR(QNOV,3)
@ 14,34 SAY STR(QDEC,3)
STCRE (QJUL+QAUG+QSEP+QOCT+QNOV+QDEC) TO MSUM
@ 16,4 SAY "A TOTAL OF "
@ 16,15 SAY STR(MSUM,3)
@ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH = '07'
@ 12,4 SAY "AUG SEP OCT NOV DEC"
@ 14,4 SAY STR(QAUG,3)
@ 14,10 SAY STR(QSEP,3)
@ 14,16 SAY STR(QCCT,3)
@ 14,22 SAY STR(QNOV,3)
@ 14,28 SAY STR(QDEC,3)
STCRE (QAUG+QSEP+QCCT+QNOV+QDEC) TO MSUM
@ 16,4 SAY "A TOTAL OF "
@ 16,15 SAY STR(MSUM,3)
@ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH = '08'
@ 12,4 SAY "SEP CCT NOV DEC"
@ 14,4 SAY STR(QSEP,3)
@ 14,10 SAY STR(QCCT,3)
@ 14,16 SAY STR(QNOV,3)
@ 14,22 SAY STR(QDEC,3)
STCRE (QSEP+QOCT+QNOV+QDEC) TO MSUM
@ 16,4 SAY "A TOTAL OF "
@ 16,15 SAY STR(MSUM,3)
@ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH = '09'
@ 12,4 SAY "OCT NOV DEC"
@ 14,4 SAY STR(QOCT,3)
@ 14,10 SAY STR(QNOV,3)
@ 14,16 SAY STR(QDEC,3)
STCRE (QOCT+QNOV+QDEC) TO MSUM
@ 16,4 SAY "A TOTAL OF "
@ 16,15 SAY STR(MSUM,3)
@ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH = '10'
@ 12,4 SAY "NOV DEC"

```



```

@ 14,4 SAY STR(ONCV,3)
@ 14,10 SAY STR(QDEC,3)
STORE (CNOV+QDEC) TO MSUM
@ 16,4 SAY "A TOTAL OF "
@ 16,15 SAY STR(MSUM,3)
@ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH='11'
@ 12,4 SAY "DEC"
@ 14,4 SAY STR(QDEC,3)
STORE (QDEC) TO MSUM
@ 16,4 SAY "A TOTAL OF "
@ 16,15 SAY STR(MSUM,3)
@ 16,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
CASE MCNTH='12'
@ 14,10 SAY "THIS IS THE LAST MONTH IN CALENDAR YEAR"
ENDCASE
DO CASE
CASE (MSUM = MLEFT)
@ 18,4 SAY "REMAINING QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY.."
@ 18,64 SAY STR(MSUM,3)
CASE (MLEFT > MSUM)
@ 18,4 SAY "YOU NEED "
STORE (MLEFT - MSUM) TO NEEDED
@ 18,13 SAY STR(NEEDED,3)
@ 18,19 SAY "MORE QUOTAS FOR THE YEAR."
CASE (MSUM > MLEFT)
@ 18,4 SAY "YOU HAVE "
STORE (MSUM - MLEFT) TO EXTRA
@ 18,13 SAY STR(EXTRA,3)
@ 18,18 SAY "EXTRA QUOTAS."
ENDCASE
@ 19,1 SAY "SELECT ONE"
@ 20,1 SAY "A...CHANGE QUOTAS ON SCRATCH PAD"
@ 21,1 SAY "B...RETURN TO STANDARD REPORT MENU"
@ 22,0
ACCEPT "SELECT OPTION ==> " TO WHATIF
IF !(WHATIF) = 'A'
DO B:QUOTAPAD
ENDIF
RELEASE MCNTH,WHATIF,EXTRA,NEEDED,MSUM,MLEFT,TOTALREC,RFLNA
RELEASE ALL LIKE Q???
RETURN

```


1. QUOTA SCRATCH PAD ROUTINE

```

* ROUTINE NAME: QUOTAFAD.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.6.2.1.0
* AUTECR: R.E. PRUIETT
* DATE: 5DEC83
* VARIABLES USED: MLEFT, MUNIT
* VARIABLES MODIFIED:
* VARIABLES CREATED: PICKMNTN, TEMPSUM, WHICHWAY, CONTINUE,
* QFEB...QDEC, EXTRA
* VARIABLES RELEASED: C???, MLEFT, TEMPSUM, WHICHWAY,
* PICKMNTN, CONTINUE, EXTRA
* FILES (OPENED/CLOSED): NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: CTASTUDY.PRG
* DESCRIPTION: THIS MODULE GIVES THE USER AN OPPORTUNITY TO
* INTER-ACTIVELY MODIFY THE REMAINING MONTHLY RANGE
* QUOTAS. THIS CAN BE DONE AN UNLIMITED NUMBER OF TIMES
* EACH TIME RETURNING TO THE ACTUAL MONTHLY ALLOCATION AS
* BASE DATA. HOWEVER, IF THE USER ELECTS TO PERMANENTLY
* REPLACE THE MONTHLY QUOTA ALLOCATION IN MEMORY, HE MAY DO
* SO BY SELECTING THE 'REPLACE CURRENT QUOTAS' OPTION.

```

```

STORE 1 TO PICKMNTN
SET CCLCN OFF

```

```

DO CASE
CASE MCNTE = '01'
DO WHILE PICKMNTN = T
DO CASE
CASE ! (MUNIT) = 'A'
RESTORE FROM A:ARNGQTAS ADDITIVE
CASE ! (MUNIT) = 'B'
RESTORE FROM A:BRNGQTAS ADDITIVE
CASE ! (MUNIT) = 'C'
RESTORE FROM A:CRNGQTAS ADDITIVE
CASE ! (MUNIT) = 'D'
RESTORE FROM A:DRNGQTAS ADDITIVE
CASE ! (MUNIT) = 'E'
RESTORE FROM A:BNRNGQTA ADDITIVE
ENDCASE
ERASE
@ 2,15 SAY "FIFTE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION "
@ 6,0 SAY "*****";
*****"
@ 8,5 SAY "FEE MAR APR MAY JUN JUL;
AUG SEP OCT NOV DEC"
@ 10,4 GET QFEB PICTURE '999'
@ 10,10 GET QMAR PICTURE '999'
@ 10,16 GET QAPR PICTURE '999'
@ 10,22 GET QMAY PICTURE '999'
@ 10,28 GET QJUN PICTURE '999'
@ 10,34 GET QJUL PICTURE '999'
@ 10,40 GET QAUG PICTURE '999'
@ 10,46 GET QSEP PICTURE '999'
@ 10,52 GET QCCT PICTURE '999'
@ 10,58 GET QNOV PICTURE '999'
@ 10,64 GET QDEC PICTURE '999'
READ
@ 12,4 SAY "A TOTAL OF "
STORE (QFEB+QMAR+QAPR+QMAY+QJUN+QJUL+;

```



```

CAUG+OSEP+QOCT+QNOV+ODEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

```

```

DC CASE
CASE (TEMPSUM = MLEFT)
@ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
@ 14,53 SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
@ 14,4 SAY "YOU NEED "
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14,13 SAY STR(NEEDED,3)
@ 14,19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
@ 14,4 SAY "YOU HAVE "
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14,13 SAY STR(EXTRA,3)
@ 14,18 SAY "EXCESS QUOTAS."
ENDCASE

```

```

@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY

```

```

DO CASE
CASE !(WHICHWAY) = 'B'
STORE F TO PICKMNTN
CASE !(WHICHWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CCNTINUE
DO CASE
CASE !(MUNIT) = 'A'
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'B'
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'C'
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'D'
SAVE TO A:DRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'E'
SAVE TO A:BNRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTN
ELSE
STORE F TO PICKMNTN
ENDIF
ENDCASE

```

ENIDC

```

CASE MCNTH = '02'
DO WHILE PICKMNTN = T
DO CASE
CASE !(MUNIT) = 'A'
RESTORE FFCM A:ARNGQTAS ADDITIVE
CASE !(MUNIT) = 'B'
RESTORE FFCM A:BRNGQTAS ADDITIVE
CASE !(MUNIT) = 'C'
RESTORE FFCM A:CRNGQTAS ADDITIVE
CASE !(MUNIT) = 'D'
RESTORE FFCM A:DRNGQTAS ADDITIVE
CASE !(MUNIT) = 'E'
RESTORE FFCM A:BNRNGQTA ADDITIVE
ENDCASE

```



```

ERASE
@ 2,15 SAY "FIFTE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELCW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YCU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION "
@ 6,0 SAY "*****";
*****"
@ 8,5 SAY "      MAR   APR   MAY   JUN   JUL;
AUG   SEP   OCT   NOV   DEC"
@ 10,10 GET QMAR PICTURE '999'
@ 10,16 GET QAPR PICTURE '999'
@ 10,22 GET QMAY PICTURE '999'
@ 10,28 GET QJUN PICTURE '999'
@ 10,34 GET QJUL PICTURE '999'
@ 10,40 GET QAUG PICTURE '999'
@ 10,46 GET QSEP PICTURE '999'
@ 10,52 GET QOCT PICTURE '999'
@ 10,58 GET QNOV PICTURE '999'
@ 10,64 GET QDEC PICTURE '999'
READ

@ 12,4 SAY "A TOTAL OF "
STORE (QMAR+QAPR+QMAY+QJUN+QJUL+;
CAUG+QSEP+QOCT+QNOV+QDEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

DO CASE
CASE (TEMPSUM = MLEFT)
@ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
@ 14,53 SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
@ 14,4 SAY "YOU NEED "
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14,13 SAY STR(NEEDED,3)
@ 14,19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
@ 14,4 SAY "YOU HAVE "
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14,13 SAY STR(EXTRA,3)
@ 14,18 SAY "EXCESS QUOTAS."
ENDCASE

@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL;
QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY

DO CASE
CASE !(WHICHWAY) = 'B'
STORE F TO PICKMNTN
CASE !(WHICHWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CCNTINUE
DO CASE
CASE !(MUNIT) = 'A'
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'B'
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'C'
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'D'
SAVE TO A:DRNGQTAS ALL LIKE Q???

```



```

        CASE !(MUNIT) = 'E'
            SAVE TO A:BNRNGQTA ALL LIKE Q???
        ENDCASE
        STORE F TO PICKMNTH
    ELSE
        STORE F TO PICKMNTH
    ENDIF
ENDCASE

```

ENDDO

CASE MCNTH = '03'

DO WHILE PICKMNTH = T

```

DO CASE
    CASE !(MUNIT) = 'A'
        RESTORE FFCM A:ARNGQTAS ADDITIVE
    CASE !(MUNIT) = 'B'
        RESTORE FFCM A:BRNGQTAS ADDITIVE
    CASE !(MUNIT) = 'C'
        RESTORE FFCM A:CRNGQTAS ADDITIVE
    CASE !(MUNIT) = 'D'
        RESTORE FFCM A:DRNGQTAS ADDITIVE
    CASE !(MUNIT) = 'E'
        RESTORE FFCM A:BNRNGQTA ADDITIVE

```

ENDCASE

FRASE

```

@ 2,15 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION "
@ 6,0 SAY "*****";
*****"
@ 8,5 SAY "      APR    MAY    JUN    JUL;
AUG    SEP    CCT    NCV    DEC"
@ 10,16 GET CAPR PICTURE '999'
@ 10,22 GET CMAY PICTURE '999'
@ 10,28 GET CJUN PICTURE '999'
@ 10,34 GET CJUL PICTURE '999'
@ 10,40 GET CAUG PICTURE '999'
@ 10,46 GET CSEP PICTURE '999'
@ 10,52 GET CCCT PICTURE '999'
@ 10,58 GET CNOV PICTURE '999'
@ 10,64 GET CDEC PICTURE '999'
FEAC

```

```

@ 12,4 SAY "A TOTAL OF "
STORE (QAPR+CMAY+QJUN+QJUL+
CAUG+QSEP+QOCT+CNOV+CDEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

```

DO CASE

```

    CASE (TEMPSUM = MLEFT)
        @ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY.:"
        @ 14,53 SAY STR(TEMPSUM,3)
    CASE (MLEFT > TEMPSUM)
        @ 14,4 SAY "YOU NEED "
        STORE (MLEFT - TEMPSUM) TO NEEDED
        @ 14,13 SAY STR(NEEDED,3)
        @ 14,19 SAY "MORE QUOTAS."
    CASE (TEMPSUM > MLEFT)
        @ 14,4 SAY "YOU HAVE "
        STORE (TEMPSUM - MLEFT) TO EXTRA
        @ 14,13 SAY STR(EXTRA,3)
        @ 14,18 SAY "EXCESS QUOTAS."
    ENDCASE

```



```

@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL;
QUCTAS"
ACCEPT "SELECT CNE ==> " TO WHICHWAY

```

```

DO CASE
CASE ! (WHICHWAY) = 'B'
    STORE F TO PICKMNTN
CASE ! (WHICHWAY) = 'C'
    ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
    ORIGINAL SET OF QUCTAS (Y/N)?" TO CONTINUE
    IF CONTINUE
        DO CASE
        CASE ! (MUNIT) = 'A'
            SAVE TO A:ARNGQTAS ALL LIKE Q???
        CASE ! (MUNIT) = 'B'
            SAVE TO A:BRNGQTAS ALL LIKE Q???
        CASE ! (MUNIT) = 'C'
            SAVE TO A:CRNGQTAS ALL LIKE Q???
        CASE ! (MUNIT) = 'D'
            SAVE TO A:DRNGQTAS ALL LIKE Q???
        CASE ! (MUNIT) = 'E'
            SAVE TO A:BNRNGQTA ALL LIKE Q???
        ENDCASE
        STORE F TO PICKMNTN
    ELSE
        STORE F TO PICKMNTN
    ENDIF
ENDCASE

```

ENDDDC

```

CASE MCNTF = '04'
DO WHILE PICKMNTN = T
DO CASE
CASE ! (MUNIT) = 'A'
    RESTORE FROM A:ARNGQTAS ADDITIVE
CASE ! (MUNIT) = 'B'
    RESTORE FROM A:BRNGQTAS ADDITIVE
CASE ! (MUNIT) = 'C'
    RESTORE FROM A:CRNGQTAS ADDITIVE
CASE ! (MUNIT) = 'D'
    RESTORE FROM A:DRNGQTAS ADDITIVE
CASE ! (MUNIT) = 'E'
    RESTORE FROM A:BNRNGQTA ADDITIVE
ENDCASE
ERASE
@ 2,15 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION "
@ 6,0 SAY "*****";
*****
@ 8,5 SAY "          MAY    JUN    JUL;
AUG    SEP    OCT    NOV    DEC"
@ 10,22 GET CMAY PICTURE '999'
@ 10,28 GET CJUN PICTURE '999'
@ 10,34 GET CJUL PICTURE '999'
@ 10,40 GET CAUG PICTURE '999'
@ 10,46 GET CSEP PICTURE '999'
@ 10,52 GET COCT PICTURE '999'
@ 10,58 GET CNCV PICTURE '999'
@ 10,64 GET QDEC PICTURE '999'
READ

```



```

@ 12,4 SAY "A TOTAL OF "
STORE (QMAY+CJUN+QJUL+
CAUG+OSEP+OOCT+QNOV+QDEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

```

```

DO CASE
CASE {TEMPSUM = MLEFT}
@ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
@ 14,53 SAY STR(TEMPSUM,3)
CASE {MLEFT > TEMPSUM}
@ 14,4 SAY "YOU NEED "
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14,13 SAY STR(NEEDED,3)
@ 14,19 SAY "MORE QUOTAS."
CASE {TEMPSUM > MLEFT}
@ 14,4 SAY "YOU HAVE "
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14,13 SAY STR(EXTRA,3)
@ 14,18 SAY "EXCESS QUOTAS."
ENDCASE

```

```

@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL;
QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY

```

```

DO CASE
CASE {(WHICHWAY) = 'B'}
STORE F TO PICKMNTH
CASE {(WHICHWAY) = 'C'}
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CONTINUE
DO CASE
CASE {(MUNIT) = 'A'}
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE {(MUNIT) = 'B'}
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE {(MUNIT) = 'C'}
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE {(MUNIT) = 'D'}
SAVE TO A:DRNGQTAS ALL LIKE Q???
CASE {(MUNIT) = 'E'}
SAVE TO A:BNRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
ENDCASE

```

ENDDC

```

CASE MCNTH = '05'
DO WHILE PICKMNTH = T
DO CASE
CASE {(MUNIT) = 'A'}
RESTORE FROM A:ARNGQTAS ADDITIVE
CASE {(MUNIT) = 'B'}
RESTORE FROM A:BRNGQTAS ADDITIVE
CASE {(MUNIT) = 'C'}
RESTORE FROM A:CRNGQTAS ADDITIVE
CASE {(MUNIT) = 'D'}
RESTORE FROM A:DRNGQTAS ADDITIVE

```



```

CASE !(MUNIT) = 'E'
  RESTORE FROM A:BNRNGQTA ADDITIVE
ENDCASE
ERASE
@ 2,15 SAY "FIFTE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION "
@ 6,0 SAY "*****";
*****"
@ 8,5 SAY "
AUG SEP CCT NOV DEC" JUN JUL;
@ 10,28 GET CJUN PICTURE '999'
@ 10,34 GET CJUL PICTURE '999'
@ 10,40 GET CAUG PICTURE '999'
@ 10,46 GET CSEP PICTURE '999'
@ 10,52 GET CCT PICTURE '999'
@ 10,58 GET CNOV PICTURE '999'
@ 10,64 GET CDEC PICTURE '999'
READ
@ 12,4 SAY "A TOTAL OF "
STORE (CJUN+CJUL+
QAUG+CSEP+COCT+CNOV+CDEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

DC CASE
CASE (TEMPSUM = MLEFT)
@ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
@ 14,53 SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
@ 14,4 SAY "YOU NEED "
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14,13 SAY STR(NEEDED,3)
@ 14,19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
@ 14,4 SAY "YOU HAVE "
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14,13 SAY STR(EXTRA,3)
@ 14,18 SAY "EXCESS QUOTAS."
ENDCASE

@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY

DC CASE
CASE !(WHICHWAY) = 'B'
  STORE F TO PICKMNTN
CASE !(WHICHWAY) = 'C'
  ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
  ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
  IF CONTINUE
    DO CASE
      CASE !(MUNIT) = 'A'
        SAVE TO A:ARNGQTA ALL LIKE Q???
      CASE !(MUNIT) = 'B'
        SAVE TO A:BRNGQTA ALL LIKE Q???
      CASE !(MUNIT) = 'C'
        SAVE TO A:CRNGQTA ALL LIKE Q???
      CASE !(MUNIT) = 'D'
        SAVE TO A:DRNGQTA ALL LIKE Q???
      CASE !(MUNIT) = 'E'

```



```

                SAVE TC A:BNRNGQTA ALL LIKE Q???
            ENDCASE
            STORE F TO PICKMNTH
        ELSE
            STORE F TO PICKMNTH
        ENDIF
    ENDCASE

ENDDO

CASE MCNTH = '06'
    LC WHILE PICKMNTH = T
    DO CASE
        CASE ! (MUNIT) = 'A'
            RESTORE FFCM A:ARNGQTAS ADDITIVE
        CASE ! (MUNIT) = 'B'
            RESTORE FFCM A:BRNGQTAS ADDITIVE
        CASE ! (MUNIT) = 'C'
            RESTORE FFCM A:CRNGQTAS ADDITIVE
        CASE ! (MUNIT) = 'D'
            RESTORE FFCM A:DRNGQTAS ADDITIVE
        CASE ! (MUNIT) = 'E'
            RESTORE FFCM A:BNRNGQTA ADDITIVE
    ENDCASE
    ERASE
    @ 2,15 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
    @ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
    CALCULATIONS AND WILL BE SAVED"
    @ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
    CURRENT QUOTAS' OPTION "
    @ 6,0 SAY "*****";
    *****"
    @ 8,5 SAY "
    JUL;
    AUG  SEP   CCT   NOV   DEC"
    @ 10,34 GET CJUL PICTURE '999'
    @ 10,40 GET CAUG PICTURE '999'
    @ 10,46 GET CSEP PICTURE '999'
    @ 10,52 GET CCCT PICTURE '999'
    @ 10,58 GET CNCV PICTURE '999'
    @ 10,64 GET QDEC PICTURE '999'
    READ

    @ 12,4 SAY "A TOTAL OF "
    STORE (QJUL+;
    CAUG+QSEP+QOCT+QNOV+QDEC) TO TEMPSUM
    @ 12,15 SAY STR(TEMPSUM,3)
    @ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

    DO CASE
        CASE (TEMPSUM = MLEFT)
            @ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
            LEFT TO QUALIFY..."
            @ 14,53 SAY STR(TEMPSUM,3)
        CASE (MLEFT > TEMPSUM)
            @ 14,4 SAY "YOU NEED "
            STORE (MLEFT - TEMPSUM) TO NEEDED
            @ 14,13 SAY STR(NEEDED,3)
            @ 14,19 SAY "MORE QUOTAS."
        CASE (TEMPSUM > MLEFT)
            @ 14,4 SAY "YOU HAVE "
            STORE (TEMPSUM - MLEFT) TO EXTRA
            @ 14,13 SAY STR(EXTRA,3)
            @ 14,18 SAY "EXCESS QUOTAS."
    ENDCASE

    @ 15,24 SAY "SELECT ONE OPTION"
    @ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
    @ 17,18 SAY "E...RETURN TO STANDARD REPORT;
    SELECTION MENU"

```



```
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY
```

```
DO CASE
CASE !(WHICHWAY) = 'B'
STORE F TO PICKMNTH
CASE !(WHICHWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CONTINUE
DO CASE
CASE !(MUNIT) = 'A'
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'B'
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'C'
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'D'
SAVE TO A:DRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'E'
SAVE TO A:BNRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
ENICASE
```

```
ENDDO
```

```
CASE MCNTE = '07'
DO WHILE PICKMNTH = T
DO CASE
CASE !(MUNIT) = 'A'
RESTORE FROM A:ARNGQTAS ADDITIVE
CASE !(MUNIT) = 'B'
RESTORE FROM A:BRNGQTAS ADDITIVE
CASE !(MUNIT) = 'C'
RESTORE FROM A:CRNGQTAS ADDITIVE
CASE !(MUNIT) = 'D'
RESTORE FROM A:DRNGQTAS ADDITIVE
CASE !(MUNIT) = 'E'
RESTORE FROM A:BNRNGQTA ADDITIVE
ENDCASE
ERASE
@ 2,15 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION "
@ 6,0 SAY "*****";
*****"
@ 8,5 SAY "
AUG SEP CCT NOV DEC;
@ 10,40 GET CAUG PICTURE '999'
@ 10,46 GET CSEP PICTURE '999'
@ 10,52 GET CCCT PICTURE '999'
@ 10,58 GET CNOV PICTURE '999'
@ 10,64 GET CDEC PICTURE '999'
FEAL

@ 12,4 SAY "A TOTAL OF "
STORE (QAUG+CSEP+QOCT+CNOV+QDEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
```

```
DO CASE
CASE (TEMPSUM = MLEFT)
@ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
```



```

LEFT TO QUALIFY..."
@ 14,53 SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
@ 14,4 SAY "YOU NEED "
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14,13 SAY STR(NEEDED,3)
@ 14,19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
@ 14,4 SAY "YOU HAVE "
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14,13 SAY STR(EXTRA,3)
@ 14,18 SAY "EXCESS QUOTAS."
ENDCASE

```

```

@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;"
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY

```

```

DO CASE
CASE !(WHICHWAY) = 'B'
STORE F TO PICKMNTN
CASE !(WHICHWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CONTINUE
DO CASE
CASE !(MUNIT) = 'A'
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'B'
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'C'
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'D'
SAVE TO A:DRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'E'
SAVE TO A:BNRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTN
ELSE
STORE F TO PICKMNTN
ENDIF
ENICASE

```

ENDDC

```

CASE MCNTE = '08'
DO WHILE PICKMNTN = T
IO CASE
CASE !(MUNIT) = 'A'
RESTORE FROM A:ARNGQTAS ADDITIVE
CASE !(MUNIT) = 'B'
RESTORE FROM A:BRNGQTAS ADDITIVE
CASE !(MUNIT) = 'C'
RESTORE FROM A:CRNGQTAS ADDITIVE
CASE !(MUNIT) = 'D'
RESTORE FROM A:DRNGQTAS ADDITIVE
CASE !(MUNIT) = 'E'
RESTORE FROM A:BNRNGQTA ADDITIVE
ENDCASE

```

```

FRASE
@ 2,15 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION "
@ 6,0 SAY "*****";

```


*****"

```
@ 8,5 SAY "
SEP OCT NCV DEC" ;
@ 10,46 GET CSEP PICTURE '999'
@ 10,52 GET CCCT PICTURE '999'
@ 10,58 GET CNCV PICTURE '999'
@ 10,64 GET CDEC PICTURE '999'
READ
```

```
@ 12,4 SAY "A TOTAL OF "
STCRÉ (QSEP+CCCT+QNCV+CDEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."
```

```
DO CASE
CASE (TEMPSUM = MLEFT)
@ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
@ 14,53 SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
@ 14,4 SAY "YOU NEED "
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14,13 SAY STR(NEEDED,3)
@ 14,19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
@ 14,4 SAY "YOU HAVE "
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14,13 SAY STR(EXTRA,3)
@ 14,18 SAY "EXCESS QUOTAS."
ENDCASE
```

```
@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY
```

```
DO CASE
CASE !(WHICHWAY) = 'B'
STORE F TO PICKMNTN
CASE !(WHICHWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CONTINUE
DO CASE
CASE !(MUNIT) = 'A'
SAVE TO A:ARNGOTAS ALL LIKE Q???
CASE !(MUNIT) = 'B'
SAVE TO A:BRNGOTAS ALL LIKE Q???
CASE !(MUNIT) = 'C'
SAVE TO A:CRNGOTAS ALL LIKE Q???
CASE !(MUNIT) = 'D'
SAVE TO A:DRNGOTAS ALL LIKE Q???
CASE !(MUNIT) = 'E'
SAVE TO A:BNRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTN
ELSE
STORE F TO PICKMNTN
ENDIF
ENICASE
```

ENDDO

```
CASE MCNTN = '09'
DO WHILE PICKMNTN = T
IO CASE
CASE !(MUNIT) = 'A'
```



```

        RESTORE FROM A:ARNGQTAS ADDITIVE
CASE ! (MUNIT) = 'B'
        RESTORE FROM A:BRNGQTAS ADDITIVE
CASE ! (MUNIT) = 'C'
        RESTORE FROM A:CRNGQTAS ADDITIVE
CASE ! (MUNIT) = 'D'
        RESTORE FROM A:DRNGQTAS ADDITIVE
CASE ! (MUNIT) = 'E'
        RESTORE FROM A:BNFNGQTA ADDITIVE
ENDCASE
FRASE
@ 2,15 SAY "FIFTE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION "
@ 6,0 SAY "*****";
*****"
@ 8,5 SAY "          ;
CCT NOV IEC"
@ 10,52 GET (CCT PICTURE '999'
@ 10,58 GET (NOV PICTURE '999'
@ 10,64 GET (DEC PICTURE '999'
FEAD

@ 12,4 SAY "A TOTAL OF "
STORE (QOCT+CNOV+ODEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

DO CASE
CASE (TEMPSUM = MLEFT)
@ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
@ 14,53 SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
@ 14,4 SAY "YOU NEED "
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14,13 SAY STR(NEEDED,3)
@ 14,19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
@ 14,4 SAY "YOU HAVE "
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14,13 SAY STR(EXTRA,3)
@ 14,18 SAY "EXCESS QUOTAS."
ENDCASE

@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY
DO CASE
CASE ! (WHICHWAY) = 'B'
STORE F TO PICKMENU
CASE ! (WHICHWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y,N)?" TO CONTINUE
IF CC CONTINUE
CASE ! (MUNIT) = 'A'
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE ! (MUNIT) = 'B'
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE ! (MUNIT) = 'C'
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE ! (MUNIT) = 'D'
SAVE TO A:DRNGQTAS ALL LIKE Q???

```



```

        CASE !(MUNIT) = 'E'
            SAVE TO A:BNRNGQTA ALL LIKE Q???
        ENDCASE
        STORE F TO PICKMNTN
    ELSE
        STORE F TO PICKMNTN
    ENDIF
ENDCASE

ENDDO

CASE MCNTH = '10'
DO WHILE PICKMNTN = T
DO CASE
CASE !(MUNIT) = 'A'
    RESTORE FFCM A:ARNGQTAS ADDITIVE
CASE !(MUNIT) = 'B'
    RESTORE FFCM A:BRNGQTAS ADDITIVE
CASE !(MUNIT) = 'C'
    RESTORE FFCM A:CRNGQTAS ADDITIVE
CASE !(MUNIT) = 'D'
    RESTORE FFCM A:DRNGQTAS ADDITIVE
CASE !(MUNIT) = 'E'
    RESTORE FFCM A:BNRNGQTA ADDITIVE
ENDCASE
FRASE
@ 2,15 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BEICW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION "
@ 6,0 SAY "*****";
*****"
@ 8,5 SAY "
;
@ 10,58 GET CNCV PICTURE '999'
@ 10,64 GET QDEC PICTURE '999'
READ

@ 12,4 SAY "A TOTAL OF "
STORE (QNOV+QDEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

DO CASE
CASE (TEMPSUM = MLEFT)
    @ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
    @ 14,53 SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
    @ 14,4 SAY "YOU NEED "
    STORE (MLEFT - TEMPSUM) TO NEEDED
    @ 14,13 SAY STR(NEEDED,3)
    @ 14,19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
    @ 14,4 SAY "YOU HAVE "
    STORE (TEMPSUM - MLEFT) TO EXTRA
    @ 14,13 SAY STR(EXTRA,3)
    @ 14,18 SAY "EXCESS QUOTAS."
ENDCASE

@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL;
QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY

```



```

DC CASE
CASE !(WHICEWAY) = 'B'
STORE F TO PICKMNTH
CASE !(WHICHWAY) = 'C'
ACCEPT "DC YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CCNTINUE
DO CASE
CASE !(MUNIT) = 'A'
SAVE TO A:ARNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'B'
SAVE TO A:BRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'C'
SAVE TO A:CRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'D'
SAVE TO A:DRNGQTAS ALL LIKE Q???
CASE !(MUNIT) = 'E'
SAVE TO A:BNRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
ENLCASE

ENDDC

CASE MCNTH = '11'
DO WHILE PICKMNTH = T
DO CASE
CASE !(MUNIT) = 'A'
RESTORE FROM A:ARNGQTAS ADDITIVE
CASE !(MUNIT) = 'B'
RESTORE FROM A:BRNGQTAS ADDITIVE
CASE !(MUNIT) = 'C'
RESTORE FROM A:CRNGQTAS ADDITIVE
CASE !(MUNIT) = 'D'
RESTORE FROM A:DRNGQTAS ADDITIVE
CASE !(MUNIT) = 'E'
RESTORE FROM A:BNRNGQTA ADDITIVE
ENDCASE
ERASE
@ 2,15 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR;
CALCULATIONS AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE;
CURRENT QUOTAS' OPTION "
@ 6,0 SAY "*****";
*****"
@ 8,5 SAY ;
REC"
@ 10,64 GET CDEC PICTURE '999'
READ

@ 12,4 SAY "A TOTAL OF "
STORE (QFEB+CMAR+QAPR+OMAY+QJUN+QJUL+;
CAUG+QSEP+QOCT+QNOV+CDEC) TO TEMPSUM
@ 12,15 SAY STR(TEMPSUM,3)
@ 12,19 SAY "QUOTAS REMAIN FOR THE CURRENT YEAR."

DO CASE
CASE (TEMPSUM = MLEFT)
@ 14,4 SAY "QUOTAS EQUAL NUMBER OF MARINES;
LEFT TO QUALIFY..."
@ 14,53 SAY STR(TEMPSUM,3)
CASE (MLEFT > TEMPSUM)
@ 14,4 SAY "YOU NEED "
STORE (MLEFT - TEMPSUM) TO NEEDED
@ 14,13 SAY STR(NEEDED,3)

```



```

@ 14,19 SAY "MORE QUOTAS."
CASE (TEMPSUM > MLEFT)
@ 14,4 SAY "YOU HAVE "
STORE (TEMPSUM - MLEFT) TO EXTRA
@ 14,13 SAY STR(EXTRA,3)
@ 14,18 SAY "EXCESS QUOTAS."
ENDCASE

```

```

@ 15,24 SAY "SELECT ONE OPTION"
@ 16,18 SAY "A...START AGAIN WITH ORIGINAL QUOTAS"
@ 17,18 SAY "E...RETURN TO STANDARD REPORT;"
SELECTION MENU"
@ 18,18 SAY "C...PERMANENTLY REPLACE ORIGINAL QUOTAS"
ACCEPT "SELECT ONE ==> " TO WHICHWAY

```

```

DO CASE
CASE !(WHICHWAY) = 'B'
STORE F TO PICKMNTH
CASE !(WHICHWAY) = 'C'
ACCEPT "DO YOU WANT TO PERMANENTLY REPLACE THE;
ORIGINAL SET OF QUOTAS (Y/N)?" TO CONTINUE
IF CONTINUE
DO CASE
CASE !(MUNIT) = 'A'
SAVE TO A:ARNGOTAS ALL LIKE Q???
CASE !(MUNIT) = 'B'
SAVE TO A:BRNGOTAS ALL LIKE Q???
CASE !(MUNIT) = 'C'
SAVE TO A:CRNGOTAS ALL LIKE Q???
CASE !(MUNIT) = 'D'
SAVE TO A:DRNGOTAS ALL LIKE Q???
CASE !(MUNIT) = 'E'
SAVE TO A:BNRNGQTA ALL LIKE Q???
ENDCASE
STORE F TO PICKMNTH
ELSE
STORE F TO PICKMNTH
ENDIF
ENDCASE

```

ENDDC

```

CASE MCNTH = '12'
ERASE
@ 2,15 SAY "RIFLE RANGE QUOTA SCRATCH PAD"
@ 4,5 SAY "SPACE BELOW THIS LINE IS FOR CALCULATIONS;"
AND WILL BE SAVED"
@ 5,5 SAY "ONLY IF YOU SPECIFY THE 'REPLACE CURRENT;"
QUOTAS' OPTION "
@ 6,0 SAY "*****";
*****"
@ 8,5 SAY "THIS IS THE LAST MONTH OF THE CURRENT;"
YEAR."
ENDCASE

```

ENDCASE

```

SET CCLCN CN
RELEASE ALL LIKE Q???
RELEASE MLEFT,TEMPSUM,WHICHWAY,PICKMNTH,CONTINUE
RETURN

```


M. MARKSMANSHIP ROSTER ROUTINE

```
* ROUTINE NAME: MKSRCST.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.7.0
* AUTHCR: R.E. PRUIETT
* DATE: 28DEC83
* VARIABLEs USED: MUNIT, MPLAT
* TEMP FILES MODIFIED: NONE
* VARIABLEs CREATED: NONE
* VARIABLEs RELEASED: MUNIT, MPLAT
* FILES OPENED/CLOSED: PERS, QUAL (ALL FILES CLOSED)
* TEMP FILES CREATED: MRKSDATA (CREATED AND DELETED)
* USING SUBROUTINES: SDRPT.PRG
* DESCRIPTION: THIS ROUTINE CREATES A TEMPORARY FILE WHICH
* INCLUDES NAME, RFIQUAL, RFLDATE, PSTQUAL, PSTDATE FOR
* EACH MEMBER IN THE SELECTED UNIT. A STANDARD REPORT
* FORMAT (B:MRKSRT) IS USED TO DISPLAY THE "MARKSMANSHIP
* ROSTER REPORT".
*
```

```
USE PERS
SELECT SECONDARY
USE QUAL
JOIN TO MRKSDATA FOR F.SSN = S.SSN FIELD;
NAME, RFIQUAL, RFLDATE, PSTQUAL, PSTDATE
USE MRKSDATA
ERASE
DO CASE
CASE ! (MUNIT) = 'A'
REPORT FORM B:MRKSRT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'A'
REPORT FORM B:MRKSRT FOR COMPANY = ONECO
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'B'
REPORT FORM B:MRKSRT FOR COMPANY = ONECO .AND.;
FIATCCN = ONEPIT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'C'
REPORT FORM B:MRKSRT FOR COMPANY = ONECO .AND.;
FIATCCN = TWOPIIT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'D'
REPORT FORM B:MRKSRT FOR COMPANY = ONECO .AND.;
FIATCCN = THREEEIT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'E'
REPORT FORM B:MRKSRT FOR COMPANY = ONECO .AND.;
FIATCCN = FOURFIT
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'A'
REPORT FORM B:MRKSRT FOR COMPANY = TWOCO
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'B'
REPORT FORM B:MRKSRT FOR COMPANY = TWOCO .AND.;
FIATCCN = ONEPIT
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'C'
REPORT FORM B:MRKSRT FOR COMPANY = TWOCO .AND.;
FIATCCN = TWOPIIT
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'D'
REPORT FORM B:MRKSRT FOR COMPANY = TWOCO .AND.;
FIATCCN = THREEEIT
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'E'
REPORT FORM B:MRKSRT FOR COMPANY = TWOCO .AND.;
FIATCCN = FOURFIT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'A'
REPORT FORM B:MRKSRT FOR COMPANY = THREECO
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'B'
REPORT FORM B:MRKSRT FOR COMPANY = THREECO .AND.;
FIATCCN = ONEPIT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'C'
REPORT FORM B:MRKSRT FOR COMPANY = THREECO .AND.;
FIATCCN = TWOPIIT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'D'
REPORT FORM B:MRKSRT FOR COMPANY = THREECO .AND.;
```



```

      FLATCCN = THREEFLT
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'E'
      REEORT FORM B:MRKSRPT FOR COMPANY = THREECO .AND.;
      FLATCCN = FOURFLT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'A'
      REEORT FORM B:MRKSRPT FOR COMPANY = FOURCO
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'B'
      REEORT FORM B:MRKSRPT FOR COMPANY = FOURCO .AND.;
      FLATCCN = CNEFLT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'C'
      REEORT FORM B:MRKSRPT FOR COMPANY = FOURCO .AND.;
      FLATCCN = TWOFLT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'D'
      REEORT FORM B:MRKSRPT FOR COMPANY = FOURCO .AND.;
      FLATCCN = THREEFLT
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'E'
      REEORT FORM B:MRKSRPT FOR COMPANY = FOURCO .AND.;
      FLATCCN = FOURFLT
ENDCASE
USE
RELEASE MUNIT, MELAT
DELETE FILE MRKSDATA
RETURN

```


N. MCS STATUS ROUTINE

```
* ROUTINE NAME: MOSSTAT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.8.0
* AUTECR: R.E. PRUIETT
* DATE: 6NOV83
* VARIABLES USED: ONFCO, TWOCO, THREECO, FOURCO, ONEFLT,
* TWCFLT, THREEFLT, FCURFLT, MUNIT, MPLAT, UNITNAME,
* PLATNAME
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: COUNT, MOS, RK, PROCEED
* VARIABLES RELEASED: PROCEED, ROW, RK, COUNT, UNITNAME,
* PLATNAME
* FILES OPENED/CLOSED: A:PERS INDEX A:MOSNDX (ALL FILES
* CLCSED)
* TEMP FILES CREATED: NONE
* USING ROUTINES: STDRPT.PRG
* DESCRIPTION: THIS ROUTINE PRODUCES THE "MOS STATUS
* REPORT". EACH TYPE OF MOS IN THE SPECIFIED UNIT IS
* COUNTED AND TOTALLED, REGARDLESS OF THE TYPE OR NUMBER OF
* DIFFERENT MOS'S IN THE PARTICULAR UNIT. THE RESULTS ARE
* THEN DISPLAYED BY MCS AND BY RANK.
```

```
*
ERASE
GO CVFTUNIT
USE A:PERS INDEX A:MCSNDX
@ 0,20 SAY "MOS STATUS REPORT"
IF !(MUNIT)='E'
@ 0,38 SAY "BATTALION"
ENDIF
IF !(MUNIT)<>'E'
@ 0,38 SAY UNITNAME
ENDIF
IF !(MPLAT)<>'E'.AND.!(MUNIT)<>'E'
@ 0,46 SAY PLATNAME
ENDIF
@ 2,10 SAY "MOS"
@ 2,30 SAY "RANK"
@ 2,55 SAY "TOTAL"
STORE 4 TO ROW
DO WHILE .NOT.EOF
SICRE PRIMEMOS TO MOS
SICRE RANK TO RK
SICRE 0 TO COUNT
DC CASE
CASE !(MUNIT)='E'
DO WHILE PRIMEMOS=MOS .AND. RANK=RK .AND. .NOT.;
EOF
STORE COUNT+1 TO COUNT
SKIP
ENDDO
CASE !(MPLAT)='E'
DO WHILE PRIMEMOS=MOS .AND. RANK=RK .AND. .NOT.;
ECF
IF COMPANY=UNITNAME
STORE COUNT+1 TO COUNT
ENDIF
SKIP
ENDDO
OTHERWISE
DO WHILE PRIMEMOS=MOS .AND. RANK=RK .AND. .NOT.;
EOF
IF COMPANY=UNITNAME .AND. PLATOON=PLATNAME
STORE COUNT+1 TO COUNT
ENDIF
SKIP
ENDDO
```



```

ENICASE
IF CCUNT > 0
  @ FOW,10 SAY MCS
  @ FOW,30 SAY RK
  @ FOW,50 SAY CCUNT
  STCRE ROW+2 TO FOW
ENDIF
IF ROW > 20
  @ 23,0
  ACCEPT "PRESS 'ENTER' TC CONTINUE" TO PROCEED
  ERASE
  STCRE 4 TO ROW
  @ 0,20 SAY "MOS STATUS REPORT"
  IF !(MUNIT)='E'
    @ 0,38 SAY "BATTALION"
  ENDIF
  IF !(MUNIT)<> 'E'
    @ 0,38 SAY UNITNAME
  ENDIF
  IF !(PLAT)<> 'E'.AND. !(MUNIT)<> 'E'
    @ 0,46 SAY PLATNAME
  ENDIF
  @ 2,10 SAY "MOS"
  @ 2,30 SAY "RANK"
  @ 2,55 SAY "TOTAL"
ENDIF
ENDDC
RELEASE PROCEED,ROW,MCS,RK,COUNT,UNITNAME,PLATNAME
USE
RETURN

```


C. MCS FOSTER ROUTINE

```
* ROUTINE NAME: MOSRCST.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.9.0
* AUTHOR: R. E. FRUETT
* DATE: 6NOV83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO,
FOURCO, ONEPLT, TWOPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: MUNIT, MPLAT
* FILES (OPENED/CLOSED): A: PERS INDEX A: MOSNDX
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTION: THIS IS A CONTROL ROUTINE WHICH CREATES THE
PERSONNEL ROSTER FOR THE UNIT SPECIFIED BY THE USER'S
RESPONSE TO THE SLCTUNIT MENU (5.1.0).
```

```
ERASE
USE A: PERS INDEX A: MCSNDX
DO CASE
CASE ! (MUNIT) = 'E'
REPORT FORM B: MCSRPT
CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'E'
REPORT FORM B: MCSRPT FOR COMPANY = ONECO
CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'A'
REPORT FORM B: MCSRPT FOR COMPANY = ONECO .AND.;
FIATCCN = ONEPLT
CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'B'
REPORT FORM B: MCSRPT FOR COMPANY = ONECO .AND.;
FIATCCN = TWOPLT
CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'C'
REPORT FORM B: MCSRPT FOR COMPANY = ONECO .AND.;
FIATCCN = THREEPLT
CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'D'
REPORT FORM B: MCSRPT FOR COMPANY = ONECO .AND.;
FIATCCN = FOURPLT
CASE ! (MUNIT) = 'E'.AND. ! (MPLAT) = 'E'
REPORT FORM B: MCSRPT FOR COMPANY = TWOCO
CASE ! (MUNIT) = 'E'.AND. ! (MPLAT) = 'A'
REPORT FORM B: MCSRPT FOR COMPANY = TWOCO .AND.;
FIATCCN = ONEPLT
CASE ! (MUNIT) = 'E'.AND. ! (MPLAT) = 'B'
REPORT FORM B: MCSRPT FOR COMPANY = TWOCO .AND.;
FIATCCN = TWOPLT
CASE ! (MUNIT) = 'E'.AND. ! (MPLAT) = 'C'
REPORT FORM B: MCSRPT FOR COMPANY = TWOCO .AND.;
FIATCCN = THREEPLT
CASE ! (MUNIT) = 'E'.AND. ! (MPLAT) = 'D'
REPORT FORM B: MCSRPT FOR COMPANY = TWOCO .AND.;
FIATCCN = FOURPLT
CASE ! (MUNIT) = 'C'.AND. ! (MPLAT) = 'E'
REPORT FORM B: MCSRPT FOR COMPANY = THREECO
CASE ! (MUNIT) = 'C'.AND. ! (MPLAT) = 'A'
REPORT FORM B: MCSRPT FOR COMPANY = THREECO .AND.;
FIATCCN = ONEPLT
CASE ! (MUNIT) = 'C'.AND. ! (MPLAT) = 'B'
REPORT FORM B: MCSRPT FOR COMPANY = THREECO .AND.;
FIATCCN = TWOPLT
CASE ! (MUNIT) = 'C'.AND. ! (MPLAT) = 'C'
REPORT FORM B: MCSRPT FOR COMPANY = THREECO .AND.;
FIATCCN = THREEPLT
CASE ! (MUNIT) = 'C'.AND. ! (MPLAT) = 'D'
REPORT FORM B: MCSRPT FOR COMPANY = THREECO .AND.;
FIATCCN = FOURPLT
CASE ! (MUNIT) = 'D'.AND. ! (MPLAT) = 'E'
REPORT FORM B: MCSRPT FOR COMPANY = FOURCO
CASE ! (MUNIT) = 'D'.AND. ! (MPLAT) = 'A'
```



```

      REPORT FORM B:MCSRPT FOR COMPANY = FOURCO .AND.;
      FLATCCN = ONEFIT
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
      REPORT FORM B:MCSRPT FOR COMPANY = FOURCO .AND.;
      FLATCCN = TWOFIT
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
      REPORT FORM B:MCSRPT FOR COMPANY = FOURCO .AND.;
      FLATCCN = THREEFIT
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
      REPORT FORM B:MCSRPT FOR COMPANY = FOURCO .AND.;
      FLATCCN = FOURFIT
ENDCASE
USE
RELEASE MUNIT, MPLAT
RETURN

```


F. PERSONNEL DATA ROUTINE

```
* ROUTINE NAME: PERSIATA.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.10.0
* AUTHOR: R. E. FRUETT
* DATE: 21NOV83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO,
* FOURCO, ONEPLT, TWOPLT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: COUNT, PROCEED
* VARIABLES RELEASED: COUNT, PROCEED
* FILES (OPENED/CLOSED): A: PERS INDEX A:ALPHPERS
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTION: THIS IS A CONTROL ROUTINE WHICH CREATES THE
* PERSONNEL ROSTER FOR THE UNIT SPECIFIED BY THE USER'S
* RESPONSE TO THE SLCTUNIT MENU(5.1.0).
```

```
ERASE
USE A:PERS INDEX A:ALPHPERS
GOTO TOP
STORE 0 TO COUNT
DO WHILE .NOT. ECF
DO CASE
CASE !(MUNIT) = 'E'
DO PERSRPT
STORE COUNT + 1 TO COUNT
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'E'
IF COMPANY = ONECO
DO PERSRPT
STORE CCOUNT + 1 TO CCOUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'A'
IF COMPANY = ONECO .AND. PLATOON = ONEPLT
DO PERSRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'B'
IF COMPANY = ONECO .AND. PLATOON = TWOPLT
DO PERSRPT
STORE CCOUNT + 1 TO CCOUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'C'
IF COMPANY = ONECO .AND. PLATOON = THREEPLT
DO PERSRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'D'
IF COMPANY = ONECO .AND. PLATOON = FOURPLT
DO PERSRPT
STORE CCOUNT + 1 TO CCOUNT
ENDIF
CASE !(MUNIT) = 'E'.AND. !(MPLAT) = 'E'
IF COMPANY = TWOCO
DO PERSRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'E'.AND. !(MPLAT) = 'A'
IF COMPANY = TWOCO .AND. PLATOON = ONEPLT
DO PERSRPT
STORE CCOUNT + 1 TO CCOUNT
ENDIF
CASE !(MUNIT) = 'E'.AND. !(MPLAT) = 'B'
IF COMPANY = TWOCO .AND. PLATOON = TWOPLT
DO PERSRPT
STORE COUNT + 1 TO COUNT
ENDIF
```



```

CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = TWCCO .AND. PLATOON = THREEPLT
    DO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  IF COMPANY = TWCCO .AND. PLATOON = FOURPLT
    DO PERSRPT
    STORE CCUNT + 1 TO CCUNT
  ENDF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
  IF COMPANY = THREECO
    DO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  IF COMPANY = THREECO .AND. PLATOON = ONEPLT
    DO PERSRPT
    STORE CCUNT + 1 TO CCUNT
  ENDF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  IF COMPANY = THREECO .AND. PLATOON = TWOPLT
    DO PERSRPT
    STORE COUNT + 1 TO CCUNT
  ENDF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  IF COMPANY = THREECO .AND. PLATOON = THREEPLT
    DO PERSRPT
    STORE CCUNT + 1 TO CCUNT
  ENDF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
  IF COMPANY = THREECO .AND. PLATOON = FOURPLT
    DO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
  IF COMPANY = FCURCO
    DO PERSRPT
    STORE CCUNT + 1 TO CCUNT
  ENDF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
  IF COMPANY = FCURCO .AND. PLATOON = ONEPLT
    DO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
  IF COMPANY = FCURCO .AND. PLATOON = TWOPLT
    DO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = FCURCO .AND. PLATOON = THREEPLT
    DO PERSRPT
    STORE COUNT + 1 TO COUNT
  ENDF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  IF COMPANY = FCURCO .AND. PLATOON = FOURPLT
    DO PERSRPT
    STORE CCUNT + 1 TO CCUNT
  ENDF
ENDCASE
IF CCUNT = 4
  @ 22,0
  ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
  STORE 0 TO COUNT
  ERASE
ENDIF
SKIP
ENDDC

```


RELEASE COUNT, PROCEED
USE
RETURN

Q. PERSONNEL DATA FORMAT ROUTINE

```
* ROUTINE NAME: PERSFET.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.10.1.0
* DATE: 21NOV83
* VARIABLES USED: COUNT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: NONE
* FILES (OPENED/CLOSED): NONE
* USING SUBROUTINES: PERSDATA.FRG
* DESCRIPTION: THIS IS A REPORT FORMAT COMMAND FILE WHICH
* DISPLAYS THE FORMAT FOR THE PERSONAL DATA REPORT AND GETS
* THE FIELD VARIABLES FROM THE PERS.DBF WHICH IS OPENED AND
* CLOSED IN PERSDATA.FRG.
```

```
*
2 COUNT * 6,0 SAY "NAME "
2 $,5 SAY NAME
2 $,42 SAY " SSN "
2 $,47 SAY SSN
2 $,1,0 SAY "RANK "
2 $,5 SAY RANK
2 $,20 SAY " PRIMARY MOS "
2 $,35 SAY PRIMEMOS
2 $,42 SAY " SECCNDAFY MOS "
2 $,70 SAY SEC MOS
2 $,1,0 SAY "COMEANY "
2 $,8 SAY CCMPANY
2 $,20 SAY " PLATOON "
2 $,30 SAY PLATOON
2 $,42 SAY " JOIN DATE "
2 $,53 SAY JOINDATE
2 $,62 SAY " EAS "
2 $,67 SAY EAS
2 $,1,0 SAY "BIRTHDATE"
2 $,10 SAY ERTHDATE
2 $,20 SAY " HEIGHT"
2 $,28 SAY FEIGHT
2 $,42 SAY " WEIGHT"
2 $,50 SAY WEIGHT
2 $,62 SAY " COMMENTS:"
2 $,73 SAY COMMENT
RETURN
```


E. EST STATUS ROUTINE

```
* ROUTINE NAME: ESTSTAT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSICN: 5.11.0
* AUTECR: R. E. PRUIETT
* DATE: 22DEC83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOOC, THREECC,
* FOURCO, ONEPLT, TWOIT, THREEFLT, FOURPLT, ZZTOTAL, ZZCOC,
* ZZHIS, ZZCOD, ZZINT, ZZAID, ZZUNI, ZZNBC, ZZMKS, ZZTAC,
* ZZFFT1, ZZPFT2
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: PROCEED
* VARIABLES RELEASED: ZZ??????
* FILES (OPENED/CLOSED): B:EST, A:PERS (ALL FILES CLOSED)
* TEMP FILES CREATED: TEMP (CREATED AND DELETED)
* USING SUBROUTINES: SDRPT.PRG
* DESCRIPTION: THIS ROUTINE CREATES A TEMPORARY FILE THAT
* INCLUDES THE ALL ESSENTIAL SUBJECT AND PFT FIELDS FROM
* THE EST.DEF FOR MEMBERS IN A USER SPECIFIED UNIT. FROM
* THIS FILE IT CCUNTS THE TCTAI NUMBER OF UNIT MEMBERS THAT
* HAVE CCOMPLETED ALL ELEMENTS IN EACH TRAINING CATEGORY.
* THE "EST STATUS REPORT" IS DISPLAYED WHICH INCLUDES
* TOTALS FOR EACH TRAINING CATEGORY AND THE PERCENTAGE
* COMPLETED FOR EACH TRAINING CATEGORY.
```

ERASE

DO CASE

CASE !(MUNIT) = 'E'

USE B:EST

DO CNTTEST

CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'E'

SELECT PRIMARY

USE A:PERS

SELECT SECCNDAFY

USE B:EST

SELECT PRIMARY

JOIN TO TEMP FOR (P.COMPANY=ONECO .AND. P.SSN=SSN);

FIELDS COC,HIS,COD,INT,AID,UNI,PFT1CLSS,PFT2CLSS,

NEC,MKS,TAC

USE TEMP

DO CNTTEST

DELETE FILE TEMP

CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'A'

SELECT PRIMARY

USE A:PERS

SELECT SECCNDAFY

USE B:EST

SELECT PRIMARY

JOIN TO TEMP FOR (P.COMPANY =ONECO .AND. P.PIATOCN =;

CNEPLT .AND. P.SSN=SSN) FIELDS COC,HIS,COD,INT,AID,;

UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC

USE TEMP

DO CNTTEST

DELETE FILE TEMP

CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'B'

SELECT PRIMARY

USE A:PERS

SELECT SECCNDAFY

USE B:EST

SELECT PRIMARY

JOIN TO TEMP FOR (P.COMPANY =ONECO .AND. P.PIATOCN;

=TWOPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,INT,;

AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC

USE TEMP

DO CNTTEST

DELETE FILE TEMP

CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'C'


```

SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JCIN TO TEMP FCF (P.CCMFANY = ONECO .AND. P.PLATOON;
= THREEPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,;
INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'A' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JCIN TO TEMP FCF (P.CCMFANY = ONECO .AND. P.PLATOON;
= FCURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,INT,;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JCIN TO TEMP FCF (P.CCMFANY=TWOCO .AND. P.SSN=SSN);
FIELDS COC,HIS,COD,INT,AID,UNI,PFT1CLSS,;
PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JCIN TO TEMP FCF (P.CCMFANY = TWOCO .AND. P.PLATOON =;
CNEPLT .AND. P.SSN=SSN) FIELDS COC,HIS,COD,INT,AID,;
UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JCIN TO TEMP FCF (P.CCMFANY = TWOCO .AND. P.PLATOON;
= TWOPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,INT,;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JCIN TO TEMP FCF (P.CCMFANY = TWOCO .AND. P.PLATOON;
= THREEPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP

```



```

DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY = TWOCO .AND. P.PLATOON =;
FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,INT,;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY = THREECO .AND. P.SSN=;
S.SSN) FIELDS COC,HIS,COD,INT,AID,UNI,PFT1CLSS,;
PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY = THREECO .AND. P.PLATOON;
= CNEFLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,INT,;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY=THREECO .AND. P.PLATOON;
= TWOPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,INT,;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY = THREECO .AND. P.PLATOON;
= THREEPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,INT,;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCF (P.CCMFANY=THREECO .AND. P.PLATOON;

```



```

= FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,;
HIS,COD,INT,AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'E'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDAFY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =FOURCO .AND.;
P.SSN=SSN) FIELDS COC,HIS,COD,INT,AID,UNI,PFT1CLSS,;
PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'A'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDAFY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =FOURCO .AND. P.PLATCCN;
=ONEPLT .AND. CNEPLT .AND. P.SSN=S.SSN) FIELDS COC,;
HIS,COD,INT,AID,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'B'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDAFY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =FOURCO .AND. P.PLATCCN;
=TWOPIT .AND. P.SSN=S.SSN) FIELDS COC,HIS,COD,INT,;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'C'
SELECT PRIMARY
USE A:PERS
SELECT SECCNDAFY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.COMPANY =FOURCO .AND. P.PLATCCN;
=THREEPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,;
INT,AID,UNI,PFT1CLSS,;
PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'D'
SELECT PRIMARY
USE A:PERS
SELECT SECONDARY
USE B:EST
SELECT PRIMARY
JOIN TO TEMP FCR (P.CCOMPANY =FOURCO .AND. P.PLATOON;
=FOURPLT .AND. P.SSN=S.SSN) FIELDS COC,HIS,CCD,INT,;
AID,UNI,PFT1CLSS,PFT2CLSS,NBC,MKS,TAC
USE TEMP
DO CNTTEST
DELETE FILE TEMP
ENDCASE
@ 0,25 SAY "EST STATUS REPORT"
@ 2,0 SAY "CODE CF CCNDUCT..."

```



```

@ 2,25 SAY (ZZTOTAL-ZZCOC)/ZZTOTAL*100 USING '999'
@ 2,30 SAY "PERCENT CCMPLETED ..."
```

Time	Command	Text
@ 2,52	SAY	ZZTOTAL-ZZCOC USING '999'
@ 4,00	SAY	"MARINE CORES HISTORY..."
@ 4,25	SAY	(ZZTOTAL-ZZHIS)/ZZTOTAL*100 USING '999'
@ 4,30	SAY	"PERCENT CCMPLETED ..."
@ 4,52	SAY	ZZTOTAL-ZZHIS USING '999'
@ 6,00	SAY	"CLOSE ORDEF DRILL..."
@ 6,25	SAY	(ZZTOTAL-ZZCOD)/ZZTOTAL*100 USING '999'
@ 6,30	SAY	"PERCENT CCMPLETED ..."
@ 6,52	SAY	ZZTOTAL-ZZCOD USING '999'
@ 8,00	SAY	"INTERIOR GUARD..."
@ 8,25	SAY	(ZZTOTAL-ZZINT)/ZZTOTAL*100 USING '999'
@ 8,30	SAY	"PERCENT CCMPLETED ..."
@ 8,52	SAY	ZZTOTAL-ZZINT USING '999'
@ 10,00	SAY	"FIRST AID..."
@ 10,25	SAY	(ZZTOTAL-ZZAID)/ZZTOTAL*100 USING '999'
@ 10,30	SAY	"PERCENT CCMPLETED ..."
@ 10,52	SAY	ZZTOTAL-ZZAID USING '999'
@ 12,00	SAY	"EQUIPMENT/UNIFORMS..."
@ 12,25	SAY	(ZZTOTAL-ZZUNI)/ZZTOTAL*100 USING '999'
@ 12,30	SAY	"PERCENT CCMPLETED ..."
@ 12,52	SAY	ZZTOTAL-ZZUNI USING '999'
@ 14,00	SAY	"NBC..."
@ 14,25	SAY	(ZZTOTAL-ZZNBC)/ZZTOTAL*100 USING '999'
@ 14,30	SAY	"PERCENT CCMPLETED ..."
@ 14,52	SAY	ZZTOTAL-ZZNBC USING '999'
@ 16,00	SAY	"MARKSMANSHIP..."
@ 16,25	SAY	(ZZTOTAL-ZZMKS)/ZZTOTAL*100 USING '999'
@ 16,30	SAY	"PERCENT CCMPLETED ..."
@ 16,52	SAY	ZZTOTAL-ZZMKS USING '999'
@ 18,00	SAY	"INDIVID IAC MEASURES..."
@ 18,25	SAY	(ZZTOTAL-ZZTAC)/ZZTOTAL*100 USING '999'
@ 18,30	SAY	"PERCENT CCMPLETED ..."
@ 18,52	SAY	ZZTOTAL-ZZTAC USING '999'
@ 20,00	SAY	"PFT1..."
@ 20,24	SAY	((ZZTOTAL-ZZPFT1)/ZZTOTAL)*100 USING '999'
@ 20,31	SAY	"PERCENT CCMPLETED PFT ONE..."
@ 20,62	SAY	(ZZTOTAL-ZZPFT1) USING '999'
@ 22,00	SAY	"PFT2..."
@ 22,24	SAY	((ZZTOTAL-ZZPFT2)/ZZTOTAL)*100 USING '999'
@ 22,31	SAY	"PERCENT CCMPLETED PFT TWO..."
@ 22,62	SAY	(ZZTOTAL-ZZPFT2) USING '999'

```

ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
RELEASE ALL LIKE ZZ??????
USE
RETURN

```


S. EST COUNT ROUTINE

```
* ROUTINE NAME: CNTEST.PRG
* MODULE NAME: STANDARD REPCRT GENERATOR
* VERSION: 5.11.1.0
* AUTHOR: R.E. PRUIETT
* DATE: 12DEC83
* VARIABLES USED: ZZTCTAL, ZZCCC, ZZHIS, ZZCOD, ZZINT,
* ZZAID, ZZUNI, ZZNBC, ZZMKS, ZZTAC, ZZPFT1, ZZPFT2
* VARIABLES MODIFIED:
* VARIABLES CREATED: NONE
* VARIABLES RELEASED: NONE
* FILES CPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: ESTSTAT.PRG
* DESCRIPTION: THIS IS A SIMPLE ROUTINE THAT COUNTS THE
* TOTAL TRAINING CATEGORIES THAT INCLUDE A "*" (I.E. COUNTS
* THE NUMBER OF TRAINING CATEGORIES THAT CURRENTLY INCLUDE
* ONE OR MORE TRAINING ELEMENTS THAT HAVE NOT BEEN
* COMPLETED.
```

```
*
COUNT TO ZZTOTAL
COUNT FOR '*'$(COC) TO ZZCOC
COUNT FOR '*'$(HIS) TC ZZHIS
COUNT FOR '*'$(COD) TC ZZCOD
COUNT FOR '*'$(INT) TC ZZINT
COUNT FOR '*'$(AID) TC ZZAID
COUNT FOR '*'$(UNI) TC ZZUNI
COUNT FOR '*'$(NBC) TC ZZNBC
COUNT FOR '*'$(MKS) TO ZZMKS
COUNT FOR '*'$(TAC) TC ZZTAC
COUNT FOR '*'$(PFT1CISS) TO ZZPFT1
COUNT FOR '*'$(PFT2CISS) TO ZZPFT2
RETURN
```


T. EST ROSTER ROUTINE

```
* ROUTINE NAME: ESTRCST.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSION: 5.12.0
* AUTHCR: R. E. PRUIETT
* DATE: 20DEC83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECO,
* FOURCO, CNEPLT, TWCLPT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: COUNT, PROCEED
* VARIABLES RELEASED: COUNT, FFOCEED
* FILES (OPENED/CLOSED): A: PERS INDEX A: ALHPERS, E: EST
* INDEX E: ESTSSN (ALL FILES ARE CLOSED).
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: SDRPT.PRG
* DESCRIPTION: THIS IS A CONTROL ROUTINE WHICH CREATES THE
* PERSONNEL ROSTER FOR THE UNIT SPECIFIED BY THE USER'S
* RESPONSE TO THE SLCUNIT MENU (5.1.0).
```

```
ERASE
SELECT PRIMARY
USE A: PERS INDEX A: ALHPERS
SELECT SECONDARY
USE B: EST INDEX B: ESTSSN
SELECT PRIMARY
GOTO TOP
STORE 0 TO COUNT
DO WHILE .NCT. EOF
DO CASE
CASE ! (MUNIT) = 'E'
DO ESTRPT
STORE COUNT + 1 TO COUNT
CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'E'
IF COMPANY = ONECO
DO ESTRPT
STORE CCOUNT + 1 TO CCOUNT
ENDIF
CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'A'
IF COMPANY = ONECO .AND. PLATOON = ONEPLT
DO ESTRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'B'
IF COMPANY = ONECO .AND. PLATOON = TWOPLT
DO ESTRPT
STORE CCOUNT + 1 TO CCOUNT
ENDIF
CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'C'
IF COMPANY = ONECO .AND. PLATOON = THREEPLT
DO ESTRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE ! (MUNIT) = 'A'.AND. ! (MPLAT) = 'D'
IF COMPANY = ONECO .AND. PLATOON = FOURPLT
DO ESTRPT
STORE CCOUNT + 1 TO CCOUNT
ENDIF
CASE ! (MUNIT) = 'E'.AND. ! (MPLAT) = 'E'
IF COMPANY = TWOCO
DO ESTRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE ! (MUNIT) = 'E'.AND. ! (MPLAT) = 'A'
IF COMPANY = TWOCO .AND. PLATOON = ONEPLT
DO ESTRPT
STORE CCOUNT + 1 TO CCOUNT
ENDIF
```



```

CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'B'
  IF COMPANY = TWOCO .AND. PLATOON = TWOPLT
    DO ESTRPT
    STCRE CCUNT + 1 TO CCUNT
  ENDIF
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'C'
  IF COMPANY = TWOCO .AND. PLATOON = THREEPLT
    DO ESTRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE ! (MUNIT) = 'E' .AND. ! (MPLAT) = 'D'
  IF COMPANY = TWOCO .AND. PLATOON = FOURPLT
    DO ESTRPT
    STORE CCUNT + 1 TO CCUNT
  ENDIF
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'E'
  IF COMPANY = TERECCO
    DO ESTRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'A'
  IF COMPANY = THREECO .AND. PLATOON = ONEPLT
    DO ESTRPT
    STCRE CCUNT + 1 TO CCUNT
  ENDIF
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'B'
  IF COMPANY = TFREECO .AND. PLATOON = TWOPLT
    DO ESTRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'C'
  IF COMPANY = THREECO .AND. PLATOON = THREEPLT
    DO ESTRPT
    STCRE CCUNT + 1 TO CCUNT
  ENDIF
CASE ! (MUNIT) = 'C' .AND. ! (MPLAT) = 'D'
  IF COMPANY = TERECCO .AND. PLATOON = FOURPLT
    DO ESTRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE ! (MUNIT) = 'D' .AND. ! (MPLAT) = 'E'
  IF COMPANY = FCURCO
    DO ESTRPT
    STCRE CCUNT + 1 TO CCUNT
  ENDIF
CASE ! (MUNIT) = 'D' .AND. ! (MPLAT) = 'A'
  IF COMPANY = FCURCO .AND. PLATOON = ONEPLT
    DO ESTRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE ! (MUNIT) = 'D' .AND. ! (MPLAT) = 'B'
  IF COMPANY = FCURCO .AND. PLATOON = TWOPLT
    DO ESTRPT
    STCRE COUNT + 1 TO COUNT
  ENDIF
CASE ! (MUNIT) = 'D' .AND. ! (MPLAT) = 'C'
  IF COMPANY = FCURCO .AND. PLATOON = THREEPLT
    DO ESTRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE ! (MUNIT) = 'D' .AND. ! (MPLAT) = 'D'
  IF COMPANY = FCURCO .AND. PLATOON = FOURPLT
    DO ESTRPT
    STCRE CCUNT + 1 TO CCUNT
  ENDIF
ENDCASE
IF CCUNT = 4
  @ 22,C
  ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED

```



```
STORE 0 TO COUNT  
ERASE  
ENDIF  
SELECT PRIMARY  
SKIP  
ENDDC  
RELEASE COUNT, PROCEED  
USE  
RETURN
```


U. EST REPORT FORMAT ROUTINE

```

* ROUTINE NAME: ESTRTT.PRG
* MODULE NAME: STANDARD REPORT GENERATOR
* VERSICN: 5.12.1.0
* AUTFOR: R.E. PRUIETT
* DATE: 14NOV83
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: ROW, MSSN
* VARIABLES RELEASED: FOW
* FILES CPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: ESTROST.PRG
* DESCRIPTION: THIS ROUTINE USES. FORMATS AND
* DISPLAYS THE EST RCSTER REPORT.
STORE (CCUNT * 6) TO FCW
@ ROW,0 SAY NAME
@ ROW,30 SAY RANK
@ ROW,33 SAY COMPANY
STORE SSN TC MSSN
SELECT SECONDARY
FIND 8MSSN
@ ROW + 1,0 SAY "CODE OF CONDUCT "
@ ROW + 1,16 SAY COC
@ ROW + 1,22 SAY "MARCOR HIS "
@ ROW + 1,35 SAY HIS
@ ROW + 1,43 SAY "MARKSMANSHIP "
@ ROW + 1,56 SAY MKS
@ ROW + 1,67 SAY "DRILL "
@ ROW + 1,74 SAY COD
@ ROW + 2,0 SAY "INTERIOR GUARD "
@ ROW + 2,16 SAY INT
@ ROW + 2,22 SAY "MARKSMANSHIP "
@ ROW + 2,35 SAY MKS
@ ROW + 2,43 SAY "TAC MEASURES "
@ ROW + 2,56 SAY TAC
@ ROW + 2,67 SAY "NBC "
@ ROW + 2,74 SAY NBC
@ ROW + 3,0 SAY "PFT1 "
@ ROW + 3,6 SAY PFT1CLSS
@ ROW + 3,10 SAY "PFT2 "
@ ROW + 3,16 SAY PFT2CLSS
@ ROW + 3,22 SAY "EQUIP/UNIFORM "
@ ROW + 3,36 SAY UNI
@ ROW + 3,43 SAY "FIRST AID "
@ ROW + 3,54 SAY AID
RELEASE FOW
RETURN

```


V. SWIM QUALIFICATION ROUTINE

```
* ROUTINE NAME: SWIMQUAL.PRG
* MODULE NAME: STANDARD REPCRT GENERATOR
* VERSION: 5.13.0
* AUTEOR: R. E. PRUITT
* DATE: 20DEC83
* VARIABLES USED: MUNIT, MPLAT, ONECO, TWOCO, THREECC,
* FOURCO, CNEPLT, TWCELT, THREEPLT, FOURPLT
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: PROCEED, COUNT, S1TOTAL, S2TOTAL,
* S3TOTAL, UQTOTAL
* VARIABLES RELEASED: COUNT, PROCEED, MSSN, S1TOTAL,
* S2TOTAL, S3TOTAL, UQTOTAL, RCW
* FILES (OPENED/CLOSED): A:PERS INDEX A:ALPHPERS
* TEMP FILES CREATED: NONE
* USING SUBROUTINES: STDRPT.PRG
* DESCRIPTION: THIS ROUTINE DISPLAYS THE SWIM QUAL REPORT
* FOR THE SELECTED UNIT. THE SWIM QUAL REPORT CONSIST OF
* EACH MARINES NAME, RANK, UNIT NAME, AND SWIMMING
* QUALIFICATION STATUS. AT THE END OF THE REPORT, SUMMARY
* DATA IS DISPLAYED WHICH INCLUDES THE TOTALS FOR EACH
* QUALIFICATION CATEGORY AND THE OVERALL TOTAL FOR THE
* UNIT.
```

```
ERASE
@ 12,27 SAY "SWIM QUALIFICATION REPORT"
@ 22,C
ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
ERASE
SELECT PFIMARY
USE A:PERS INDEX A:ALPHPERS
GOTO TOP
STORE 0 TO S1TOTAL
STORE 0 TO S2TOTAL
STORE 0 TO S3TOTAL
STORE 0 TO UQTOTAL
STORE 0 TO COUNT
DO WHILE .NOT. EOF
DO CASE
CASE !(MUNIT) = 'E'
DO SWIMRPT
STORE COUNT + 1 TO COUNT
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'E'
IF COMPANY = ONECO
DO SWIMRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'A'
IF COMPANY = ONECO .AND. PLATOON = ONEPLT
DO SWIMRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'B'
IF COMPANY = ONECO .AND. PLATOON = TWOPLT
DO SWIMRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'C'
IF COMPANY = ONECO .AND. PLATOON = THREEPLT
DO SWIMRPT
STORE COUNT + 1 TO COUNT
ENDIF
CASE !(MUNIT) = 'A'.AND. !(MPLAT) = 'D'
IF COMPANY = ONECO .AND. PLATOON = FOURPLT
DO SWIMRPT
STORE COUNT + 1 TO COUNT
ENDIF
```



```

CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'E'
  IF COMPANY = TWCCO
    DO SWIMRPT
    STCRE CCUNT + 1 TO CCUNT
  ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'A'
  IF COMPANY = TWCCO .AND. PLATOON = ONEPLT
    DO SWIMRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'B'
  IF COMPANY = TWCCO .AND. PLATOON = TWOPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'C'
  IF COMPANY = TWOCO .AND. PLATOON = THREEPLT
    DO SWIMRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE !(MUNIT) = 'E' .AND. !(MPLAT) = 'D'
  IF COMPANY = TWCCO .AND. PLATOON = FOURPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDIF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'E'
  IF COMPANY = THREECO
    DO SWIMRPT
    STORE CCUNT + 1 TO COUNT
  ENDIF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'A'
  IF COMPANY = THREECO .AND. PLATOON = ONEPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO CCUNT
  ENDIF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'B'
  IF COMPANY = THREECO .AND. PLATOON = TWOPLT
    DO SWIMRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'C'
  IF COMPANY = THREECO .AND. PLATOON = THREEPLT
    DO SWIMRPT
    STCRE CCUNT + 1 TO CCUNT
  ENDIF
CASE !(MUNIT) = 'C' .AND. !(MPLAT) = 'D'
  IF COMPANY = THREECO .AND. PLATOON = FOURPLT
    DO SWIMRPT
    STORE CCUNT + 1 TO COUNT
  ENDIF
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'E'
  IF COMPANY = FCURCO
    DO SWIMRPT
    STCRE CCUNT + 1 TO CCUNT
  ENDIF
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'A'
  IF COMPANY = FCURCO .AND. PLATOON = ONEPLT
    DO SWIMRPT
    STORE COUNT + 1 TO COUNT
  ENDIF
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'B'
  IF COMPANY = FCURCO .AND. PLATOON = TWOPLT
    DO SWIMRPT
    STCRE COUNT + 1 TO COUNT
  ENDIF
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'C'
  IF COMPANY = FCURCO .AND. PLATOON = THREEPLT
    DO SWIMRPT
    STORE COUNT + 1 TO COUNT
  
```



```

        ENDIF
CASE !(MUNIT) = 'D' .AND. !(MPLAT) = 'D'
    IF COMPANY = FCURCO .AND. PLATOON = FOURPLT
        DO SWIMRPT
        STCRE CCUNT + 1 TO CCUNT
    ENDIF
ENDCASE
IF CCUNT = 7
    @ 22,C
    ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
    STCRE 0 TO COUNT
    ERASE
ENDIF
SELECT PFIMARY
SKIP
ENDDO
@ 22,C
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ERASE
@ 2,20 SAY "SUMMARY DATA"
@ 4,1 SAY "SWIM QUALIFICATION"
@ 4,25 SAY "TOTAL"
@ 6,1 SAY "S1"
@ 6,25 SAY S1TOTAL USING '999'
@ 7,1 SAY "S2"
@ 7,25 SAY S2TOTAL USING '999'
@ 8,1 SAY "S3"
@ 8,25 SAY S3TOTAL USING '999'
@ 9,1 SAY "UNQUALIFIED"
@ 9,25 SAY UQTOTAL USING '999'
@ 10,25 SAY " "
@ 12,25 SAY S1TOTAL+S2TOTAL+S3TOTAL+UQTOTAL USING '9999'
@ 22,0
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
RELEASE CCUNT, PROCEED, MSSN, S1TOTAL, S3TOTAL, ;
UQTOTAL, FOW
USE
RETURN

```


W. SWIM REPORT ROUTINE

```
* ROUTINE NAME: SWIMRPT.PRG
* MODULE NAME: STANDARD REPCRT GENERATOR
* VERSICN: 5.13.1.0
* AUTHEOR: R.E. PRUIETT
* DATE: 12DEC83
* VARIABLES USED:NONE
* VARIABLES MODIFIED:NONE
* VARIABLES CREATED:ROW, MSSN, S1TOTAL, S2TOTAL, S3TCTAL,
* UQTCTAL
* VARIABLES RELEASED:NONE
* FILES CPENED/CLOSED:QUAL INDEX QUALSSN
* TEMP FILES CREATED:NONE
* USING SUBROUTINES: SWIMQUAL.PRG
* DESCRIPTION: THIS ROUTINE CCUNTS AND STORES THE NUMBER OF
* PERSONNEL IN THE SELECTED UNIT THAT ARE IN EACH SWIMMING
* QUALIFICATION CATEGCRY.
```

```
STORE (CCUNT * 3) TO ROW
@ ROW,0 SAY NAME
@ ROW,30 SAY RANK
@ ROW,33 SAY COMPANY
STORE SSN TO MSSN
SELECT SECONDARY
USE B:QUAL INDEX B:QUALSSN
FIND &MSSN
@ ROW+1,C SAY "SWIM QUAL "
@ ROW+1,10 SAY SWIMQUAL
DO CASE
  CASE SWIMQUAL='S1'
    SICRE S1TOTAL + 1 TO S1TCTAL
  CASE SWIMQUAL='S2'
    SICRE S2TOTAL + 1 TO S2TCTAL
  CASE SWIMQUAL='S3'
    SICRE S3TOTAL + 1 TO S3TCTAL
  CASE SWIMQUAL='UQ'
    SICRE UQTOTAL + 1 TO UQTCTAL
ENDCASE
RETURN
```


I. HELP ROUTINE

```
* ROUTINE NAME:  HELP.PRG
* MODULE NAME:  STANDARD REPORT GENERATOR
* VERSION: 5.14.0
* AUTEOR: R.E. PRUIETT
* DATE: 18JAN84
* VARIABLES USED: NONE
* VARIABLES MODIFIED: NONE
* VARIABLES CREATED: EELOPT, PROCEED
* VARIABLES RELEASED: NONE
* FILES OPENED/CLOSED: NONE
* TEMP FILES CREATED: NONE
* USING FOUTINES: STIRPT
* DESCRIPTION: THIS ROUTINE PROVIDES A FUNCTIONAL
* DESCRIPTION OF EACH REPORT LISTED IN THE STANDARD REPORT
* SELECTION MENU.
```

```
ERASE
@ 2,35 SAY "HELP MENU"
@ 3,35 SAY "-----"
@ 5,20 SAY "A...HELP UNIT ROSTER"
@ 6,20 SAY "B...HELP TRAINING STATUS REPORT"
@ 7,20 SAY "C...HELP TRAINING ROSTER REPORT"
@ 8,20 SAY "D...HELP INDIVIDUAL TRAINING RECORD"
@ 9,20 SAY "E...HELP MARKSMANSHIP STATUS REPORT"
@ 10,20 SAY "F...HELP MARKSMANSHIP ROSTER REPORT"
@ 11,20 SAY "G...HELP MOS STATUS REPORT"
@ 12,20 SAY "H...HELP MCS ROSTER REPORT"
@ 13,20 SAY "I...HELP PERSONAL DATA REPORT"
@ 14,20 SAY "J...HELP EST STATUS REPORT"
@ 15,20 SAY "K...HELP EST ROSTER REPORT"
@ 16,20 SAY "L...HELP SWIM QUALIFICATION REPORT"
@ 22,20
```

```
ACCEPT "SELECT OPTION =====> " TO HELPOPT
```

```
ERASE
DO CASE
```

```
  CASE !(HELPOPT) = 'A'
```

```
TEXT
```

OPTION A...UNIT ROSTER

1. At the cursor, next to "select option =====>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The UNIT ROSTER REPORT will display in alphabetic order the NAME, RANK, and PRIMARY MOS for each member of the selected unit or subunit. The screen will scroll automatically unless stopped from the terminal.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.

```
ENDTEXT
```

```
CASE !(HELPOPT) = 'B'
```

```
TEXT
```

OPTION B...TRAINING STATUS REPORT

1. At the cursor, next to "select option =====>", type and enter the letter (A-E) that corresponds to the unit desired.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. A "WORKING" signal on the screen will be shown to indicate that the computer is searching for the information you requested.

5. The TRAINING STATUS REPORT provides for all essential subjects, the PERCENT of the unit that has successfully completed each training element and the TOTAL number of unit members that have completed each training element. In addition, the report includes completion status for PFT1, PFT2, SWIMQUAL, RIFLE QUAL, and PISTOL QUAL.

6. Press the enter key to return to the STANDARD REPORT MENU.
ENDTEXT

CASE ! (HELPCT) = 'C'
TEXT

OPTION C...TRAINING ROSTER REPORT

1. At the cursor, next to "select option ==>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The TRAINING ROSTER REPORT heading will appear in the center of the screen. Press the enter key to begin the report.
ENDTEXT

2 22,0

ACCEPT "PRESS 'ENTER' KEY TO CONTINUE" TO PROCEED
ERASE
TEXT

5. The TRAINING ROSTER REPORT displays in alphabetic order each member of the selected unit or subunit and his complete training status for the current year. If a training element has been successfully completed for the current year, a letter corresponding to that training element will be shown. A "*" symbol indicates that a training element has not been completed. For example, "FIRST AID AE*H*B**CF" indicates that elements ABDEF and H have been completed and five of the eleven elements have NOT been completed (i.e., C, G, I, J and K). Four records will be displayed on the screen and you will be asked to press the enter key to scroll the report.

6. At the end of the report press the enter key to return to the STANDARD REPORT MENU.
ENDTEXT

CASE ! (HELPCT) = 'D'
TEXT

OPTION D...INDIVIDUAL TRAINING REPORT

1. At the cursor, next to "select option ==>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.
 2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).
 3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.
 4. At the end of the report press the enter key to return to the STANDARD REPORT MENU.
- ENDTEXT

CASE ! (EELPCPT) = 'E'
TEXT

OPTION E...MARKSMANSHIP STATUS REPORT

1. At the cursor, next to "select option ==>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.
 2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).
 3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.
 4. The MARKSMANSHIP STATUS REPORT will display the qualification results for the unit or subunit members that have fired for qualification during the current year. Also, the percent that has qualified in each category (eg. EX, SS, MM, UNC) and current year results for rifle and pistol will be shown. The pistol qualification results include only those members required to fire the pistol for qualification. Press the enter key to continue with the report.
- ENDTEXT
2 22,0
ACCEPT "PRESS 'ENTER' KEY TO CCNTINUE" TO PROCEED
ERASE

TEXT

5. You will be asked if you want to do a company or battalion range quota analysis. If you answer no ("N"), you will automatically return to the STANDARD REPORT MENU. The range quota analysis allows you to observe, manipulate or permanently change the rifle range quotas that are assigned to the unit for the rest of the calendar year. Range quota analysis may only be done for the battalion or for a selected company (not platoons).
6. At the top of the screen general rifle range quota information is displayed followed by the number of quotas remaining in the current calendar year. Finally, an indication whether remaining quotas are less than, greater than, or equal to the number of Marines left to qualify is shown.
7. From this point you may change the quota allocation on a scratch pad or you may return to the STANDARD REPORT MENU. The original set of quotas have not been modified.


```
ENDTEXT
2 22,0
ACCEPT "PRESS 'ENTER' KEY TO CCNTINUE" TO PROCEED
ERASE
TEXT
```

8. The scratch pad again shows the remaining quota allocations. However, you may make any temporary quota changes you desire on the scratch pad by simply overtyping the current quota allocation(s). The enter key will automatically move the cursor to the next quota, and a buzzer will sound when you reach the end of each field.

9. After you've reached the last month's quota, you will be asked to select one of the following three options:

- A...Start over on the scratch pad with the original set of quotas.
- B...Return to the STANDARD REPORT MENU
- C...PERMANENTLY REPLACE ORIGINAL QUOTAS with the set of quotas that are currently displayed on the screen. This is the only option that will actually modify the original set of quotas for the unit.

10. Option A may be repeated indefinitely. Option B or option C will return you directly to the STANDARD REPORT MENU.

```
ENDTEXT
```

```
CASE !(HELPCT) = 'F'
TEXT
```

OPTION F...MARKSMANSHIP ROSTER REPORT

1. At the cursor, next to "select option ==>", type the letter (A-F) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The MARKSMANSHIP ROSTER REPORT will display in alphabetical order the NAME, RANK, DATE OF QUALIFICATION, and QUALIFICATION RESULTS for each member of the unit or subunit. Current year results only will be displayed.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.

```
ENDTEXT
```

```
CASE !(HELPCT) = 'G'
TEXT
```

OPTION G...MOS STATUS REPORT

1. At the cursor, next to "select option ==>", type the letter (A-F) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The MCS STATUS REPORT displays the number of unit or subunit members grouped by MOS and RANK. The screen will scroll automatically unless stopped from your terminal.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.
ENDTEXT

CASE ! (HELP CPT) = 'H'
TEXT

OPTION H...MOS ROSTER REPORT

1. At the cursor, next to "select option ==>", type the letter (A-F) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The MCS ROSTER REPORT groups by primary mos and displays in alphabetic order the NAME, RANK, PRIMARY MOS AND SECONDARY MOS of all members in the unit or subunit.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.
ENDTEXT

CASE ! (HELP CPT) = 'I'
TEXT

OPTION I...PERSONAL DATA REPORT

1. At the cursor, next to "select option ==>", type the letter (A-F) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The PERSONAL DATA REPORT displays for each member of the unit or subunit all personal data including NAME, SSN, RANK, PRIMARY/SECONDARY MOS, UNIT/SUBUNIT NAME, JOIN DATE, EAS, BIRTHDATE, HEIGHT, WEIGHT and a one character comment block. If the comment block is "T" (true), a comment is contained on the individual concerned in the comment file. The screen will scroll upon pressing the enter key.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.
ENDTEXT

CASE ! (HELP CPT) = 'J'
TEXT

OPTION J...EST STATUS REPORT

1. At the cursor, next to "select option ==>", type the letter (A-F) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked

to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The EST STATUS REPORT includes the percent and total number of unit or subunit members that have completed each ESSENTIAL SUBJECT and the PFT.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.
ENDTEXT

CASE !(HELPCT)= 'K'
TEXT

OPTION K...EST ROSTER REPORT

1. At the cursor, next to "select option ==>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The EST ROSTER REPORT displays for each member of the unit or subunit all of his essential subject training and PFT results for the current year. A "*" character indicates an element that has not been completed. The screen will scroll upon pressing the enter key.

5. At the end of the report press the enter key to return to the STANDARD REPORT MENU.
ENDTEXT

CASE !(HELPCT)= 'L'
TEXT

OPTION L...SWIM QUALIFICATION REPORT

1. At the cursor, next to "select option ==>", type the letter (A-E) that corresponds to the unit desired. Enter the letter by pressing the enter key.

2. If a company was selected (A-D), you will next be asked to select a subunit. Enter the appropriate letter. If you want the report to include the entire unit selected, enter "E" (i.e., ALL OF UNIT).

3. If option "E" (i.e., BATTALION) was selected in the select unit option, you will go directly to the report and will not be asked to select a subunit.

4. The SWIM QUALIFICATION REPORT displays in alphabetic order for each member of the unit or subunit NAME, RANK, UNIT and CURRENT SWIM QUAL RESULTS.

5. At the end of the report a SUMMARY DATA TABLE is displayed which shows the number of unit or subunit members that have qualified in each category (S1, S2, S3, UNQ).

6. Press the enter key to return to the STANDARD REPORT MENU.
ENDTEXT

ENDCASE
2 23,C
ACCEPT "PRESS 'ENTER' TO CONTINUE" TO PROCEED
ENDCASE
RETURN

APPENDIX H MAINTENANCE MODULE LISTING

A. MAINTAIN CONTROL ROUTINE

```

* Routine Name: Maintain.Prg
* Module Name: Maintenance Module
* Version: 6.0
* Author: D.P. Haeusler
* Date: 30 Nov 83
* Variables Used: mdcne, mntnopt
* Variables Modified: mdone, mntnopt
* Variables Created: mdone, mntnopt
* Variables Released: mdone, mntnopt
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Cmddet
* Description: This routine produces a module which gives
* the user the option of which maintenace function to
* activate. Each function has an authorization level
* associated with and only appropriate users may activate
* certain modules.
    
```

```

STORE f TO AAdone
DO WHILE .NOT. AAdone
    ERASE
    @ 4,29 SAY "MAINTENANCE MODULE MENU"
    @ 6,24 SAY "SELECT ONE OF THE FOLLOWING OPTIONS"
    @ 8,31 SAY "V...VIEW INDIVIDUAL DATA"
    @ 9,31 SAY "U...UPDATE DATA"
    @ 10,31 SAY "C...CREATE AN ITR"
    @ 11,31 SAY "L...DELETE AN ITR"
    @ 12,31 SAY "S...SYSTEM FUNCTIONS"
    @ 13,31 SAY "H...HELP"
    @ 14,29 SAY "* Q...QUIT *"
    @ 15,0 SAY " "

    ACCEPT "                SELECT OPTION ==>" TO mntnopt

    DC CASE
    CASE !(mntnopt) = 'V'
        DO viewitr
    CASE !(mntnopt) = 'U'
        IF AAlevel < 3
            DO updtitr
        ELSE
            ERASE
            @ 8,31 SAY "ACCESS UNAUTHORIZED"
            DO pause
        ENDIF
    CASE !(mntnopt) = 'C'
        IF AAlevel < 3
            DO creatitr
        ELSE
            ERASE
            @ 8,31 SAY "ACCESS UNAUTHORIZED"
            DO pause
        ENDIF
    CASE !(mntnopt) = 'D'
        IF AAlevel = 1
            DO deletitr
    
```



```

ELSE
    ERASE
    @ 8,31 SAY "ACCESS UNAUTHORIZED"
    DO pause
ENDIF
CASE !(mntnopt) = 'S'
    IF AAlevel = 1
        DO system
    ELSE
        ERASE
        @ 8,31 SAY "ACCESS UNAUTHORIZED"
        DO pause
    ENDIF
CASE !(mntnopt) = 'H'
    STORE 2 TO HELPOPT
    DO B:HEIP2
CASE !(mntnopt) = 'Q'
    STORE t TO AAdone
    ERASE
ENDCASE
ENDDO
RELEASE mntnopt, HELPCPT
RELEASE AALCNE
RETURN

```


E. VIEW CONTROL ROUTINE

```
* Routine Name: Viewitr.prg
* Module Name: Maintenance Module
* Version: 6.1.0
* Author: D.P. Haeusler
* Date: 1 Dec 83
* Variables Used: vdcne, vfinish, vname, vcomp, vagain,
*                vssn, vend, view
* Variables Modified: same as above
* Variables Created: same as above
* Variables Released: all like v???????
* Files (opened/closed): a:pers(opened/closed)
* Temporary Files Created: none
* Using Subroutines: maintain.prg, stdrpt.prg
* Description: This routine controls the creation of the
* ITR for viewing only. It does not allow any changes to
* data.
```

```
STORE F TC VNOFIND
STORE F TC VDONE
```

```
DO WHILE .NCT. VDONE
  STCRE F TO VFINISH
  DC WHILE .NOT. VFINISH .AND. .NOT. VDONE
    FRASE
    @ 8,31 SAY "TO ICCATE AN ITR"
    @ 9,31 SAY "ENTER NAME (LAST, FIRST MI.)"
    ACCEPT " NAME ==>" TO VNAME

    STCRE !(VNAME) TO VNAME
    USE A:PERS INDEX A:ALFHPERS
    FIND &VNAME

    IF # = 0
      FRASE

      @ 8,31 SAY "INDIVIDUAL IS NOT IN THE DATA BASE"
      STCRE F TO VCOMP
      DO WHILE .NCT. VCCMP
        ACCEPT "DO YOU WANT TO TRY AGAIN (Y/N)" ;
        TO VAGAIN
        STORE !(VAGAIN) TO VAGAIN
        DO CASE
          CASE VAGAIN = 'N'
            STCRE T TC VICNE
            STORE T TO VCOMP
            STCRE T TC FINISH
          CASE VAGAIN = 'Y'
            STCRE T TC VCCMP
            STORE T TO VNOFIND
        ENDCASE
      ENDDO
    ELSE
      STORE SSN TO AASSN
      STCRE COMMENT TO STCCMMNT
      USE

      STCRE F TO VEND
      DO WHILE .NCT. VEND
        STORE T TC VEND
        DO ITRSCRN
        DO GETDATA
        @ 24,24 SAY "*** PRESS ANY KEY TO CONTINUE ***"
        SET CONSCIE OFF
        WAIT
        SET CONSCIE ON
        STORE F TC VIEW
```



```

IF STCOMMENT
  STORE F TO VCOMP
  DO WHILE .NOT. VCOMP
    ACCEPT " DO YOU WANT TO VIEW COMMENTS;
(Y/N)?" TO VAGAIN
    DO CASE
      CASE ! (VAGAIN) = 'N'
        STORE T TO VCOMP
        STORE F TO VIEW
        STORE I TO VFINISH
        STORE T TO VDONE
      CASE ! (VAGAIN) = 'Y'
        STORE T TO VCOMP
        STORE I TO VIEW
    ENDCASE
  ENDDO
  IF VIEW
    DO GETCMMNT
  ENDIF
  @ 24,0
  STORE T TO VFINISH
  STORE F TO VCOMP
  DO WHILE .NOT. VCOMP .AND. VIEW
    ACCEPT " DO YOU WANT TO VIEW ITR AGAIN;
(Y/N)?" TO VAGAIN
    DO CASE
      CASE ! (VAGAIN) = 'Y'
        STORE T TO VCOMP
        STORE F TO VEND
      CASE ! (VAGAIN) = 'N'
        STORE T TO VCOMP
        STORE T TO VEND
    ENDCASE
  ENDDO
  ENLDO
  ENLDO
  ENLDO
  ERASE
  STCRE F TO VCOMP
  DO WHILE .NOT. VCCMF .AND. .NOT. VNOFIND
    @ 11,24 SAY " TO VIEW ANOTHER ITR"
    ACCEPT " ENTER (Y/N) ==>";
    TO VAGAIN
    DO CASE
      CASE ! (VAGAIN) = 'Y'
        STORE T TO VCOMP
      CASE ! (VAGAIN) = 'N'
        STORE T TO VCOMP
        STORE T TO VDONE
    ENDCASE
  ENLDO
  ENDDC
  RELEASE ALL EXCEPT AA??????
  RETURN

```


C. IIF SCREEN FORMAT ROUTINE

```
* Routine Name: Itrscin.prg
* Module Name: Maintenance Module
* Version: 6.1.0.1
* Author: D.P. Haeusler
* Date: 1 DEC 83
* Variables Used: NONE
* Variables Modified: NONE
* Variables Created: NONE
* Variables Released: NONE
* Files (opened/closed): NONE
* Temporary Files Created: NONE
* Using Subroutines: Viewitr, Creatitr, Updtitr, Deletitr
* Description: This routine formats the screen in the
* representation of an ITR.
```

ERASE

```
@ 0,25 SAY "INDIVIDUAL TRAINING RECORD"
@ 1,8 SAY "NAME:"
@ 1,46 SAY "GRADE:"
@ 1,57 SAY "MOS:"
@ 1,66 SAY "/:"
@ 2,8 SAY "BIRTHDATE.....:"
@ 2,40 SAY "SSN.....:"
@ 3,8 SAY "CO:"
@ 3,18 SAY "PLAT:"
@ 3,33 SAY "JOIN:"
@ 3,46 SAY "EAS.:"
@ 3,58 SAY "GAS MASK...:"
@ 4,8 SAY "MIL CODE OF CONDUCT:"
@ 4,40 SAY "PFT1.....:"
@ 4,55 SAY "/:"
@ 4,61 SAY "DATE:"
@ 5,8 SAY "HISTORY OF MARCOR...:"
@ 5,40 SAY "PFT2.....:"
@ 5,55 SAY "/:"
@ 5,61 SAY "DATE:"
@ 6,8 SAY "DRILL.....:"
@ 6,40 SAY "NBC.....:"
@ 7,8 SAY "INTERIOR GUARD.....:"
@ 7,40 SAY "MARKSMANSHIP.....:"
@ 8,8 SAY "FIRST AID.....:"
@ 8,40 SAY "INDIVIDUAL TACTICAL"
@ 9,8 SAY "EQUIP & UNIFORM.....:"
@ 9,40 SAY "MEASURES.....:"
@ 11,8 SAY "SWIM QUAL.....:"
@ 11,24 SAY "DATE:"
@ 11,40 SAY "WEIGHT CONTROL.....:"
@ 12,8 SAY "RIFLE QUAL...:"
@ 12,25 SAY "/:"
@ 12,31 SAY "DATE:"
@ 12,49 SAY "WEIGHT...:"
@ 13,8 SAY "PISTOL QUAL...:"
@ 13,25 SAY "/:"
@ 13,31 SAY "DATE:"
@ 13,49 SAY "HEIGHT...:"
@ 15,8 SAY "ELECTRONIC WARFARE...:"
@ 15,40 SAY "DRUG ABUSE.....:"
@ 16,8 SAY "CLD WEATHER.....:"
@ 16,40 SAY "ALCOHOL ABUSE.....:"
@ 17,8 SAY "LAW OF WAR.....:"
@ 17,40 SAY "HUMAN RELATIONS.....:"
@ 18,8 SAY "TRAINING MOS.....:"
@ 18,40 SAY "PERSONAL AFFAIRS...:"
@ 19,8 SAY "LEADERSHIP.....:"
@ 19,40 SAY "UCMJ.....:"
```


@ 20,8 SAY "CHAR & MORAL ED.....:"
@ 21,8 SAY "CMMENTS.:"

RETURN

D. SCREEN DATA RETRIEVALROUTINE

```
* Routine Name: Getdata.prg
* Module Name: Maintenance Module
* Version: 6.1.0.2
* Author: D. P. Haeusler
* Date: 1 Dec 83
* Variables Used: vssn
* Variables Modified: NONE
* Variables Created: NONE
* Variables Released: NONE
* Files (opened/closed): a:pers(opened/closed),
*                          b:est(opened/closed),
*                          b:qual(opened/closed),
*                          b:infcmisc(opened/closed)
* Temporary Files Created: NONE
* Using Subroutines: Viewitr, Updtitr, Deletitr
* Description: This routine retrieves the information
* stored in the data base for the individual specified by
* vssn. It then arranges it in the screen format created
* by Itrscrn.prg.
```

```
STORE AASSN TO VSSN
USE A:PEFS INDEX A:PERSSN
FIND EVSSN
```

```
@ 1,13 SAY NAME
@ 1,52 SAY RANK
@ 1,61 SAY PRIMEMOS
@ 1,68 SAY SEC MOS
@ 2,23 SAY ERTHDATE
@ 2,49 SAY SSN
@ 3,11 SAY COMPANY
@ 3,23 SAY FLATOON
@ 3,38 SAY JOINDATE
@ 3,51 SAY EAS
@ 3,69 SAY GASMASK
@ 11,59 SAY WTCONT
@ 12,59 SAY WEIGHT
@ 13,59 SAY HEIGHT
@ 21,17 SAY COMMENT
```

```
USE B:EST INDEX B:ESTSSN
FIND EVSSN
```

```
@ 4,28 SAY COC
@ 4,51 SAY PFT1RAW
@ 4,57 SAY PFT1CLSS
@ 4,66 SAY PFT1DATE
@ 5,28 SAY HIS
@ 5,51 SAY PFT2RAW
@ 5,57 SAY PFT2CLSS
@ 5,66 SAY PFT2DATE
@ 6,28 SAY COD
@ 6,48 SAY NEC
@ 7,28 SAY INT
@ 7,59 SAY MKS
@ 8,28 SAY AID
@ 9,28 SAY UNI
@ 9,59 SAY TAC
```

```
USE B:QUAL INDEX B:QUALSSN
FIND EVSSN
```

```
@ 11,20 SAY SWIMQUAL
@ 11,29 SAY SWIMDATE
@ 12,21 SAY RFLSCORE
@ 12,27 SAY RFLQUAL
```


@ 12,36 SAY RFLDATE
@ 13,21 SAY PSTSCORE
@ 13,27 SAY PSTQUAL
@ 13,36 SAY PSTDATE

USE B:INFCMISC INDEX E:INFOSSN
FIND EVSSN

@ 15,29 SAY ELECWAR
@ 15,59 SAY DRUG
@ 16,29 SAY CLDWTHR
@ 16,59 SAY ALCOHOL
@ 17,29 SAY LAWWAR
@ 17,59 SAY HUMREL
@ 18,29 SAY MOS
@ 18,59 SAY PERSAFFR
@ 19,29 SAY LDRSHP
@ 19,59 SAY UCMJ
@ 20,29 SAY CHRMCREC

USE
RETURN

E. COMMENT RETRIEVAL ROUTINE

```
* Routine Name: Getcmmnt.prg
* Module Name: Maintenance Module
* Version: 6.1.1
* Author: D.P. Haesusler
* Date: 10 Jan 84
* Variables Used: none
* Variables Modified: none
* Variables Created: none
* Variables Released: none
* Files (opened/closed): a:cmmnt (o/c)
* Temporary Files Created: none
* Using Subroutines: Viewitr.prg
* Description: This routine displays the contents of the
* comments data base file.
```

```
ERASE
@ 5,33 SAY "COMMENT SECTION"
USE A:CMMNT INDEX A:CMMNTSSN
FIND &AASSN
@ 8,2 SAY INFOTXT1
@ 8,51 SAY INFOTXT2
@ 9,2 SAY INFOTXT3
@ 9,51 SAY INFOTXT4
@ 10,2 SAY INFOTXT5
@ 10,51 SAY INFOTXT6
@ 11,2 SAY INFOTXT7
@ 11,51 SAY INFOTXT8
@ 12,2 SAY INFOTXT9
@ 12,51 SAY INFOTXT0
USE
RETURN
```


F. UPDATE CONTROL ROUTINE

```
* Routine Name: Updtitr.prg
* Module Name: Maintenance Module
* Version: 6.2.0
* Author: D.P. Haeusler
* Date: 2 Dec 83
* Variables Used: aaadone, upname, upcomp, upagain, aassn
* Variables Modified: aaadone, upname, upcomp, upagain,
                    aassn
* Variables Created: aaadone, upname, upcomp, upagain,
                    aassn
* Variables Released: all like up??????, aassn, aaadone
                    aathru, aaccrr
* Files (opened/closed): a:pers (OPENED/CLOSED)
* Temporary Files Created: NONE
* Using Subroutines: Maintain.prg
* Description: This routine finds the ITR to be updated
* and controls the update routines.
```

STORE F TO AAADONE

DO WHILE .NOT. AAADONE

```
ERASE
@ 8,24 SAY "WHICH ITR IS TO BE"
@ 9,23 SAY "UPDATED OR CORRECTED?"
@ 10,23 SAY "NAME (LAST, FIRST MI.)"
ACCEPT " NAME ==>" TO UPNAME
```

```
STORE !(UPNAME) TC UPNAME
USE A:PERS INDEX a:ALPHPERS
FIND @UPNAME
```

IF # = 0.

```
ERASE
@ 8,24 SAY "INDIVIDUAL IS NOT IN THE DATA BASE"
@ 9,23 SAY "OR YOU HAVE ENTERED AN INCORRECT NAME"
```

STORE F TO UPCOMP

DO WHILE .NOT. UPCOMP

```
ACCEPT " DO YOU WANT TO TRY AGAIN (Y/N)?"
TC UPAGAIN
DO CASE
CASE !(UPAGAIN) = 'Y'
STORE T TO UPCOMP
```

```
CASE !(UPAGAIN) = 'N'
STORE T TO UPCOMP
STORE T TO AAADCNE
```

ENICASE

ENDDO

ELSE

STORE SSN TO AASSN

USE

DC ITRSCRN

IO GETDATA

DC UPDATA

STORE F TO UPCOMP

DO WHILE .NOT. UPCOMP

ERASE

ACCEPT " DO YOU WANT TO UPDATE ANOTHER ITR;

(Y/N)?" TC UPAGAIN

DO CASE

```
CASE !(UPAGAIN) = 'Y'
STORE T TO UPCOMP
```

```
CASE !(UPAGAIN) = 'N'
STORE T TO UPCOMP
```



```
STORE T TO AAADONE
ENDCASE
ENDDO
ENDIF
ENDDO
RELEASE ALL EXCEPT AA??????
RELEASE AASSN, AATHRU, AACORR, AAADONE
RETURN
```


G. DATA UPDATE ROUTINE

```
* Routine Name: Updata.prg
* Module Name: Maintenance Module
* Version: 6.2.0.1
* Author: D. P. Haeusler
* Date: 8 Dec 83
* Variables Used: aathru, aacorr, upopt, inname, inrank,
*                upccrmos, inprime, insec, inbrth, insen,
*                ericpt, incomp, inplat, injdate, ineas,
*                ingas, incode, inprt1, calcopt, wthru,
*                wdate, inpfddt1, inpfddt2, inswdt,
*                inrfltdt, inpstddt, inhisc, inpf2, indrill
*                innbc, inint, inmks, inaid, inuni,
*                intac, inswgu, inrflsc, inweight,
*                inpstsc, inhght, inelec, indrug,
*                incldwtr, inalch, inlaw, inhum, inmcs,
*                inpersaf, inldr, inucmj, inchrmor,
*                inccmnt
* Variables Modified: same as above
* Variables Created: same as above
* Variables Released: all like in??????
* Files (opened/closed): a:pers (opened/closed)
*                        b:qual (opened/closed)
*                        b:est (opened/closed)
*                        b:infcmisc (opened/closed)
* Temporary Files Created: none
* Using Subroutines: updtitr.prg
* Description: This routine prompts the user to choose a
* specific field of the ITR to update. It then locates
* that field in the data base and replaces the old data
* with new data. It also calls error checking routines
* and a calculate function.
```

STORE F TO AATHRU

```
DO WHILE .NOT. AATHRU
  STORE F TO AACORR
  DO WHILE .NOT. AACORR
    RELEASE ALL LIKE IN??????
    STORE " " TO UPOPT
    @ 22,0
    @ 23,0
    @ 22,1 SAY " OPTION: FIELD TO BE CHANGED OR 'Q'";
  TO QUIT:
    @ 23,1 SAY " OPTION ==>" GET UPOPT PICTURE "XXXXXX";
  XXXXXX XXXXX XXXXXXXXXXXXXXXX"
  READ
  STORE !(UPCPT) TO UPOPT
  DO CASE
    CASE UPOPT = 'NAME'
      STORE T TO AACORR
      USE A:PEFS INDEX A:PERSSN, A:ALPHPERS
      FIND &AASSN
      STORE NAME TO INNAME
      @ 1,12 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXX";
  XXXXXX XXXXX"
  READ
  REPLACE NAME WITH !(INNAME)
  USE

  CASE UPCPT = 'GRADE'
    STORE T TO AACORR
    USE A:PEFS INDEX A:PERSSN
    FIND &AASSN
    STORE RANK TO INRANK
    @ 1,51 GET INRANK PICTURE "XX"
  READ
```



```

STORE 1 TC ERROPT
DO ERROR
REPLACE RANK WITH !(INRANK)
USE

CASE UOPT = 'MOS'
STORE T TC AACORR
@ 23,0
@ 24,0
STORE F TO UPCORMOS
DO WHILE .NOT. UPCORMOS
@ 23,1 SAY " PRIMARY OR SECONDARY?"
ACCEPT "ENTER F OR S ==>" TO UPAGAIN
DO CASE
CASE !(UPAGAIN) = 'P'
STORE T TO UPCORMOS
USE A: PERS INDEX A: PERSSN , A: MCSNDX
FIND &AASSN
STORE PRIMEMOS TO INPRIME
@ 1,60 GET INPRIME PICTURE "9999"
READ
REPLACE PRIMEMOS WITH !(INPRIME)
USE

CASE !(UPAGAIN) = 'S'
STORE T TC UPCORMOS
USE A: PERS INDEX A: PERSSN
FIND &AASSN
STORE SECMOS TO INSEC
@ 1,67 GET INSEC PICTURE "9999"
READ
REPLACE SECMOS WITH INSEC
USE
ENDCASE
ENDDO

CASE UOPT = 'BIRTHDATE'
STORE T TC AACORR
USE A: PERS INDEX A: PERSSN
FIND &AASSN
STORE BIRTHDATE TO INBRTH
@ 2,22 GET INBRTH PICTURE "XXXXXX"
READ
STORE 2 TC ERROPT
DO ERROR
REPLACE BIRTHDATE WITH INBRTH
USE

CASE UOPT = 'SSN'
STORE T TC AACORR
USE A: PERS INDEX A: PERSSN
FIND &AASSN
STORE SSN TO INSSN
@ 2,48 GET INSSN PICTURE "999999999"
READ
REPLACE SSN WITH INSSN
USE
USE A: CMMNT INDEX A: CMMNTSSN
FIND &AASSN
REPLACE SSN WITH INSSN
USE B: EST INDEX B: ESTSSN
FIND &AASSN
REPLACE SSN WITH INSSN
USE B: QUAL INDEX B: QUALSSN
FIND &AASSN
REPLACE SSN WITH INSSN
USE B: INFOMISC INDEX B: INFOSSN
FIND &AASSN
REPLACE SSN WITH INSSN

```



```

STORE INSSN TO AASSN
USE

CASE UPOPT = 'CO'
STORE T TC AACORR
USE A:PEFS INDEX A:PERSSN
FIND &AASSN
STORE COMPANY TC INCOMP
@ 3,10 GET INCOMP PICTURE "XXXXXX"
READ
REPLACE COMPANY WITH !(INCOMP)
USE

CASE UPOPT = 'PLAT'
STORE T TC AACORR
USE A:PEFS INDEX A:PERSSN
FIND &AASSN
STORE PLAT TO INPLAT
@ 3,22 GET INPLAT PICTURE "XXXXXXXXXX"
READ
REPLACE PIATON WITH !(INPLAT)
USE

CASE UPCPT = 'JOIN'
STORE T TC AACORR
USE A:PEFS INDEX A:PERSSN
FIND &AASSN
STORE JOINDATE TO INJNDATE
@ 3,37 GET INJNDATE PICTURE "XXXXXX"
READ
STORE 3 TC ERROPT
DO ERROR
REPLACE JCINDATE WITH INJNDATE
USE

CASE UPOPT = 'EAS'
STORE T TC AACORR
USE A:PEFS INDEX A:PERSSN
FIND &AASSN
STORE EAS TO INEAS
@ 3,50 GET INEAS PICTURE "XXXXXX"
READ
REPLACE EAS WITH INEAS
USE

CASE UPOPT = 'GAS MASK'
STORE T TC AACORR
USE A:PEFS INDEX A:PERSSN
FIND &AASSN
STORE GASMASK TC INGAS
@ 3,68 GET INGAS PICTURE "X"
READ
STORE 4 TC ERROPT
DO ERROR
REPLACE GASMASK WITH !(INGAS)
USE

CASE UPOPT = 'MIL CODE OF CONDUCT'
STORE T TC AACORR
USE B:EST INDEX B:ESTSSN
FIND &AASSN
STORE COC TO INCODE
@ 4,27 GET INCODE PICTURE "XXX"
READ
REPLACE CCC WITH !(INCODE)
USE

CASE UPOPT = 'PFT1'
STORE T TC AACORR

```



```

USE B:EST INDEX E:ESTSSN
FIND &AASSN
STORE PFT1RAW TC INPFT1
@ 4,50 GET INPFT1 PICTURE "999"
READ
STORE 5 TC ERROPT
DO ERROR
REPLACE PFT1RAW WITH INPFT1
USE
STORE 1 TO CALCOPT
DO CALCUIAT

CASE UPOPT = 'DATE'
STORE F TC WTHRU
STORE T TC AACORR
DO WHILE .NOT. WTHRU
    @ 23,0
    @ 24,0
    @ 23,1 SAY " A..PFT1, B..PFT2, C..SWIM,;
D..RIFLE, E..PISTOL"
    ACCEPT "WHICH DATE IS TO BE ENTERED ==> ";
    TO WDATE
    STORE !(WDATE) TO WDATE
    DO CASE
        CASE WDATE = 'A'
            STORE T TO WTHRU
            USE B:EST INDEX B:ESTSSN
            FIND &AASSN
            STORE PFT1DATE TO INPFTDT1
            @ 4,65 GET INPFTDT1 PICTURE "XXXXXX"
            READ
            STORE 6 TC ERROPT
            LC ERROR
            REPLACE PFT1DATE WITH INPFTDT1
            USE
        CASE WDATE = 'B'
            STORE T TO WTHRU
            USE B:EST INDEX B:ESTSSN
            FIND &AASSN
            STORE PFT2DATE TO INPFTDT2
            @ 5,65 GET INPFTDT2 PICTURE "XXXXXX"
            READ
            STORE 8 TO ERROPT
            LC ERROR
            REPLACE PFT2DATE WITH INPFTDT2
            USE
        CASE WDATE = 'C'
            STORE T TO WTHRU
            USE B:QUAL INDEX B:QUALSSN
            FIND &AASSN
            STORE SWIMDATE TO INSWDT
            @ 11,28 GET INSWDT PICTURE "XXXXXX"
            READ
            STORE 10 TC ERROPT
            LC ERROR
            REPLACE SWIMDATE WITH INSWDT
            USE
        CASE WDATE = 'D'
            STORE T TO WTHRU
            USE B:QUAL INDEX B:QUALSSN
            FIND &AASSN
            STORE RFLDATE TO INRFLDT
            @ 12,35 GET INRFLDT PICTURE "XXXXXX"
            READ
            STORE 12 TO ERROPT
            LC ERROR

```



```

        REPLACE RFIDATE WITH INRFIDT
        USE

    CASE WDATE = 'E'
        STORE T TO WTHRU
        USE B:QUAL INDEX B:QUALSSN
        FIND &AASSN
        STORE PSTDAT TO INPSTD
        @ 13,35 GET INPSTD PICTURE "XXXXXX"
        READ
        STORE 15 TO ERROPT
        DO ERROR
        REPLACE PSTDAT WITH INPSTD
        USE
    ENDCASE
ENDDO

CASE UPCPT = 'HISTCRY CF MARCOR'
    STORE T TO AACORR
    USE B:EST INDEX B:ESTSSN
    FIND &AASSN
    STORE HIS TO INHIS
    @ 5,27 GET INHIS PICTURE "XXX"
    READ
    REPLACE HIS WITH INHIS
    USE

CASE UPOPT = 'PFT2'
    STORE T TO AACORR
    USE B:EST INDEX B:ESTSSN
    FIND &AASSN
    STORE PFT2RAW TO INPFT2
    @ 5,50 GET INPFT2 PICTURE "999"
    READ
    STORE 7 TO ERROPT
    DO ERROR
    REPLACE PFT2RAW WITH INPFT2
    USE
    STORE 2 TO CALCCPT
    DO CALCULAT

CASE UPCPT = 'DRILL'
    STORE T TO AACORR
    USE B:EST INDEX B:ESTSSN
    FIND &AASSN
    STORE COD TO INDRILL
    @ 6,27 GET INDRILL PICTURE "XXX"
    READ
    REPLACE COD WITH INDRILL
    USE

CASE UPOPT = 'NBC'
    STORE T TO AACORR
    USE B:EST INDEX B:ESTSSN
    FIND &AASSN
    STORE NBC TO INNBC
    @ 6,47 GET INNBC PICTURE "XXXXXX"
    READ
    REPLACE NEC WITH INNBC
    USE

CASE UPCPT = 'INTERIOF GUARD'
    STORE T TO AACORR
    USE B:EST INDEX B:ESTSSN
    FIND &AASSN
    STORE INT TO ININT
    @ 7,27 GET ININT PICTURE "XXX"
    READ
    REPLACE INT WITH ININT

```



```

USE
CASE UPOPT = 'MARKSMANSHIP'
  STORE T TC AACORR
  USE B:EST INDEX B:ESTSSN
  FIND &AASSN
  STORE MKS TO INMKS
  @ 7,58 GET INMKS PICTURE "XXXXXX"
  READ
  REPLACE MKS WITH INMKS
  USE

CASE UPOPT = 'FIRST AID'
  STORE T TO AACORR
  USE B:EST INDEX B:ESTSSN
  FIND &AASSN
  STORE AID TO INAID
  @ 8,27 GET INAID PICTURE "XXXXXXXXXXXX"
  READ
  REPLACE AID WITH INAID
  USE

CASE UPOPT = 'EQUIP & UNIFORM'
  STORE T TC AACORR
  USE B:EST INDEX B:ESTSSN
  FIND &AASSN
  STORE UNI TO INUNI
  @ 9,27 GET INUNI PICTURE "XXX"
  READ
  REPLACE UNI WITH INUNI
  USE

CASE UPCPT = 'INDIVIDUAL TACTICAL MEASURES'
  STORE T TC AACORR
  USE B:EST INDEX B:ESTSSN
  FIND &AASSN
  STORE TAC TO INTAC
  @ 9,58 GET INTAC PICTURE "XXXXX"
  READ
  REPLACE TAC WITH INTAC
  USE

CASE UPOPT = 'SWIM QUAL'
  STORE T TC AACORR
  USE B:QUAL INDEX B:QUALSSN
  FIND &AASSN
  STORE SWIMQUAL TO INSWQU
  @ 11,19 GET INSWQU PICTURE "XX"
  READ
  STORE 9 TC ERRCP
  DO ERROR
  REPLACE SWIMQUAL WITH INSWQU
  USE

CASE UPOPT = 'RIFLE QUAL'
  STORE T TC AACORR
  USE B:QUAL INDEX B:QUALSSN
  FIND &AASSN
  STORE RFLSCORE TO INRFLSC
  @ 12,20 GET INRFLSC PICTURE "999"
  READ
  STORE 11 TO ERROPT
  DO ERROR
  REPLACE RFLSCORE WITH INRFLSC
  USE
  STORE 3 TC CALCOPT
  DO CALCUIAT

CASE UPOPT = 'WEIGHT'

```



```

STORE T TC AACORR
USE A: PERS INDEX A: PERSSN
FIND &AASSN
STORE WEIGHT TO INWEIGHT
@ 12,58 GET INWEIGHT PICTURE "999"
READ
STORE 13 TO ERRCPT
DO ERROR
REPLACE WEIGHT WITH INWEIGHT
USE
STORE 5 TC CALCCPT
DO CALCUIAT

CASE UPOPT = 'PISTCL QUAL'
STORE T TC AACORR
USE B: QUAL INDEX B: QUALSSN
FIND &AASSN
STORE PSTSCORE TO INPSTSC
@ 13,20 GET INPSTSC PICTURE "999"
READ
STORE 14 TO ERROPT
DO ERROR
REPLACE PSTSCORE WITH INPSTSC
USE
STORE 4 TC CALCOPT
DO CALCUIAT

CASE UPOPT = 'HEIGHT'
STORE T TC AACORR
USE A: PERS INDEX A: PERSSN
FIND &AASSN
STORE HEIGHT TO INHGHT
@ 13,58 GET INHGHT PICTURE "99"
READ
STORE 16 TO ERRCPT
DO ERROR
REPLACE HEIGHT WITH INHGHT
USE
STORE 5 TC CALCCPT
DO CALCUIAT

CASE UPOPT = 'ELECTRONIC WARFARE'
STORE T TC AACORR
USE B: INFCMISC INDEX B: INFOSSN
FIND &AASSN
STORE ELECWAR TO INELEC
@ 15,28 GET INELEC PICTURE "XXXXXX"
READ
REPLACE ELECWAR WITH INELEC
USE

CASE UPOPT = 'DRUG ABUSE'
STORE T TC AACORR
USE B: INFCMISC INDEX B: INFOSSN
FIND &AASSN
STORE DRUG TO INDRUG
@ 15,57 GET INDRUG PICTURE "XXXXXX"
READ
REPLACE DRUG WITH INDRUG
USE

CASE UPOPT = 'CLD WEATHER'
STORE T TC AACORR
USE B: INFCMISC INDEX B: INFOSSN
FIND &AASSN
STORE CLDWTHR TO INCLDWTR
@ 16,28 GET INCLDWTR PICTURE "XXXXXX"
READ
REPLACE CLDWTHR WITH INCLDWTR

```


USE

```
CASE UPOPT = 'ALCOHOL ABUSE'
STORE T TC AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE ALCOHOL TO INALCH
@ 16,58 GET INALCH PICTURE "XXXXXX"
READ
REPLACE AICOHOL WITH INALCH
USE
```

```
CASE UPCPT = 'LAW CP WAR'
STORE T TO AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE LAWWAR TO INIAW
@ 17,28 GET INLAW PICTURE "XXXXXX"
READ
REPLACE LAWWAR WITH INLAW
USE
```

```
CASE UPOPT = 'HUMAN RELATIONS'
STORE T TC AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE HUMREL TO INHUM
@ 17,58 GET INHUM PICTURE "XXXXXX"
READ
REPLACE HUMREL WITH INHUM
USE
```

```
CASE UPOPT = 'TRAINING MOS'
STORE T TO AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE MOS TO INMOS
@ 18,27 GET INMOS PICTURE "XXXX"
READ
REPLACE MCS WITH INMOS
USE
```

```
CASE UPOPT = 'PERSONAL AFFAIRS'
STORE T TC AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE PERSAFFR TO INPERSAF
@ 18,58 GET INPERSAF PICTURE "XXXXXX"
READ
REPLACE PERSAFFR WITH INPERSAF
USE
```

```
CASE UPOPT = 'LEADERSHIP'
STORE T TC AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE LDRSHIP TO INLDR
@ 19,28 GET INLDR PICTURE "XXXXXX"
READ
REPLACE LDRSHIP WITH INLDR
USE
```

```
CASE UPOPT = 'UCMJ'
STORE T TC AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE UCMJ TO INUCMJ
@ 19,58 GET INUCMJ PICTURE "XXXXXX"
READ
```



```

REPLACE UCMJ WITH INUCMJ
USE

CASE UPOPT = 'CHAR & MORAL ED'
STORE T TC AACORR
USE B:INFCMISC INDEX B:INFOSSN
FIND &AASSN
STORE CHERMCRE TO INCHRMOR
@ 20,28 GET INCHRMOR PICTURE "XXXXXX"
READ
REPLACE CERMORED WITH INCHRMOR
USE

CASE UPOPT = 'CMMENTS'
STORE T TC AACORR
USE A:PERS INDEX A:PERSSN
FIND &AASSN
IF COMMENT
    DO WRTCMMNT
    DO ITRSCRN
    DO GETDATA
ELSE
    STORE F TO INCCMMNT
    @ 21,16 GET INCOMMNT PICTURE "X"
    READ
    REPLACE COMMENT WITH INCOMMNT
    USE
    IF INCCMMNT
        DO CRTCMNT
        DO WRTCMMNT
        DO ITRSCRN
        DO GETDATA
    ENDIF
ENDIF

CASE UPOPT = 'Q'
STORE T TC AACORR
STORE T TC AATHRU
ENICASE
ENDDC
ENDEC
RETURN

```


B. ITR CREATION CONTROL ROUTINE

```
* Routine Name: Creatitr.prg
* Module Name: Maintenance Module
* Version: 6.3.0
* Author: D.F. HAEUSLER
* Date: 2 Dec 83
* Variables Used: crccmp, aafini, cragain
* Variables Modified: same as above
* Variables Created: same as above
* Variables Released: same as above
* Files (opened/closed): NONE
* Temporary Files Created: NONE
* Using Subroutines: Maintain.prg
* Description: This routine controls the creation of new
* ITR's.
```

STORE F TO AAFINI

```
DO WHILE .NCT. AAFINI
  ERASE
  DC IIFSCRN
  DC INIATA
```

SICRE F TO CRCOMP

```
DO WHILE .NOT. CRCCMP
  ERASE
  ACCEPT "DO YOU WANT TO CREATE ANOTHER ITR (Y/N)?" ;
  TO CRAGAIN
  DO CASE
    CASE ! (CRAGAIN) = 'Y'
      STORE T TO CRCOMP
    CASE ! (CRAGAIN) = 'N'
      STORE T TO CRCOMP
      STORE T TO AAFINI
```

ENICASE

ENDDO

ENDDC

RELEASE ALL LIKE CR?????

RELEASE AAFINI

USE

RETURN

I. ITR DATA INITIALIZATION ROUTINE

```
* Routine Name: Indata.prg
* Module Name: Maintenance Module
* Version: 6.3.1
* Author: D.P. Haeusler
* Date: 2 Dec 83
* Variables Used: aadate, vssn, in*, aassn
* Variables Modified: in*, aassn
* Variables Created: in*, aassn
* Variables Released: in*, aassn
* Files (opened/closed): a:pers(0/C), b:est(0/C),
                        b:infcmisc(0/C), b:qual(0/C)
* Temporary Files Created: None
* Using Subroutines: Creatitr.prg
* Description: This routine inserts the data in the
* training record format on the screen and checks for
* errors in the data entry. The data is then inserted in
* the data base.
```

```
STORE AADATE TO INDATE
STORE "LAST, FIRST MI. " TO INNAME
STORE " " TO INRANK
STORE " " TO INPRIMEM
STORE " " TO INSECMOS
STORE "DDMMYY" TO INEETH
STORE " " TO INSSN
STORE " " TC INCCMP
STORE " " TO INPLAT
STORE "DDMMYY" TO INJNDATE
STORE "DDMMYY" TO INEAS
STORE " " TC INGAS
STORE "****" TO INCODE
STORE 000 TC INPFT1
STORE "DDMMYY" TC INPFTDT1
STORE "****" TO INHIS
STORE 000 TC INPFT2
STORE "DDMMYY" TC INPFTDT2
STORE "****" TO INDRILL
STORE "*****" TO INNEC
STORE "****" TO ININT
STORE "*****" TC INPKS
STORE "*****" TC INAID
STORE "****" TO INUNI
STORE "*****" TO INTAC
STORE "S " TO INSWQU
STORE "DDMMYY" TO INSWDT
STORE "OCO" TO INRFLSC
STORE "DDMMYY" TC INRFLDT
STORE "000" TO INWEIGHT
STORE "000" TO INPSTSC
STORE "DDMMYY" TC INPSTDT
STORE "00" TO INHGHGT
STORE "*****" TO INEIEC
STORE "*****" TO INEBUG
STORE "*****" TO INCIDWTR
STORE "*****" TO INAICH
STORE "*****" TC INIAW
STORE "*****" TO INEUM
STORE "*****" TO INMOS
STORE "*****" TO INFERSAF
STORE "*****" TO INIER
STORE "*****" TO INUCMJ
STORE "*****" TO INCHRMOR
STORE F TO INCOMMNT
STORE F TO INWTCNT
STORE " " TC INPFT1CI
STORE " " TC INPFT2CI
```



```
STORE " " TO INRFLQU
STORE " " TO INPSTQU
```

```
@ 1,12 GET INNAME PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 1,51 GET INRANK PICTURE "XX"
@ 1,60 GET INPRIME PICTURE "9999"
@ 1,67 GET INSECMOS PICTURE "9999"
@ 2,22 GET INBIRTH PICTURE "XXXXXX"
@ 2,48 GET INSSN PICTURE "9999999999"
@ 3,10 GET INCOMP PICTURE "XXXXXX"
@ 3,22 GET INPLAT PICTURE "XXXXXXXXXX"
@ 3,37 GET INJDATE PICTURE "XXXXXX"
@ 3,50 GET INEAS PICTURE "XXXXXX"
@ 3,68 GET INGAS PICTURE "X"
@ 4,27 GET INCODE PICTURE "XXX"
@ 4,50 GET INPFT1 PICTURE "999"
@ 4,65 GET INPFTDT1 PICTURE "XXXXXX"
@ 5,27 GET INHIS PICTURE "XXX"
@ 5,50 GET INPFT2 PICTURE "999"
@ 5,65 GET INPFTDT2 PICTURE "XXXXXX"
@ 6,27 GET INDRILL PICTURE "XXX"
@ 6,47 GET INNBC PICTURE "XXXXXX"
@ 7,27 GET ININT PICTURE "XXX"
@ 7,58 GET INMKS PICTURE "XXXXXX"
@ 8,27 GET INAID PICTURE "XXXXXXXXXXXX"
@ 9,27 GET INUNI PICTURE "XXX"
@ 9,58 GET INTAC PICTURE "XXXXXX"
@ 11,19 GET INSWCU PICTURE "XX"
@ 11,28 GET INSWDT PICTURE "XXXXXX"
@ 12,20 GET INRFISC PICTURE "999"
@ 12,35 GET INRFLDT PICTURE "XXXXXX"
@ 12,58 GET INWEIGHT PICTURE "999"
@ 13,20 GET INPSTSC PICTURE "999"
@ 13,35 GET INPSTD T PICTURE "XXXXXX"
@ 13,58 GET INHGHT PICTURE "99"
@ 15,28 GET INELEC PICTURE "XXXXXX"
@ 15,58 GET INDRUG PICTURE "XXXXXX"
@ 16,28 GET INCLDWTR PICTURE "XXXXXX"
@ 16,58 GET INALCH PICTURE "XXXXXX"
@ 17,28 GET INLAW PICTURE "XXXXXX"
@ 17,58 GET INHUM PICTURE "XXXXXX"
@ 18,28 GET INMOS PICTURE "XXXX"
@ 18,58 GET INPERSAF PICTURE "XXXXXX"
@ 19,28 GET INLDB PICTURE "XXXXXX"
@ 19,58 GET INUCMJ PICTURE "XXXXXX"
@ 20,28 GET INCHMOR PICTURE "XXXXXX"
@ 21,16 GET INCOMMNT PICTURE "X"
```

```
READ
STORE INSSN TO AASSN
STORE 1 TO ERROPT
DO WHILE ERROPT < 17
  @ 24,0 SAY "
  @ 24,1 SAY " ERROR CHECKING IN PROGRESS"
  DC ERFOR
  STORE (ERROPT + 1) TO ERROPT
ENDDC
IF INCOMMNT
  SAVE TO A:INFILE ALL LIKE IN?????
  RELEASE ALL LIKE IN?????
  RELEASE ALL EXCEPT AA?????
  DO CRICMMNT
  DC WRICMMNT
  RESTORE FROM A:INFILE ADDITIVE
  DO ITFSCN
@ 1,13 SAY INNAME
@ 1,52 SAY INRANK
```



```

@ 1,61 SAY INPRIMEM
@ 1,68 SAY INSECMOS
@ 2,223 SAY INBRTH
@ 2,49 SAY INSSN
@ 3,111 SAY INCOMP
@ 3,23 SAY INPLAT
@ 3,38 SAY INJNDATE
@ 3,51 SAY INEAS
@ 3,69 SAY INGAS
@ 4,28 SAY INCODE
@ 4,51 SAY INPFT1 USING "999"
@ 4,66 SAY INPFTDT1
@ 5,28 SAY INHIS
@ 5,51 SAY INPFT2 USING "999"
@ 5,66 SAY INPFTDT2
@ 6,28 SAY INDRILL
@ 6,48 SAY INNBC
@ 7,28 SAY ININT
@ 7,59 SAY INMKS
@ 8,28 SAY INAID
@ 9,28 SAY INUNI
@ 9,59 SAY INTAC
@ 11,20 SAY INSWOU
@ 11,29 SAY INSWDT
@ 12,21 SAY INRFLSC
@ 12,36 SAY INRFIDT
@ 12,59 SAY INWEIGHT
@ 13,21 SAY INPSTSC
@ 13,36 SAY INPSTDIT
@ 13,59 SAY INHGHT
@ 15,29 SAY INELEC
@ 15,59 SAY INDRUG
@ 16,29 SAY INCLDWTR
@ 16,59 SAY INALCH
@ 17,29 SAY INLAW
@ 17,59 SAY INHUM
@ 18,29 SAY INMOS
@ 18,59 SAY INPERSAF
@ 19,29 SAY INLDR
@ 19,59 SAY INUCMJ
@ 20,29 SAY INCHRMOR
@ 21,17 SAY INCOMMNT
ENDIF
@ 24,1 SAY "STORING DATA IN THE SYSTEM ON DISK"
USE A:PERS INDEX A:PERSSN, A:ALHPERS, A:MOSNDX
APPEND BLANK
REPLACE NAME WITH !(INNAME)
REPLACE SSN WITH INSSN
REPLACE RANK WITH INRANK, PRIMEMOS WITH INPRIMEM
REPLACE SECMOS WITH INSECMOS, COMPANY WITH !(INCOMP)
REPLACE PLATOON WITH !(INPLAT), JOINDATE WITH INJNDATE
REPLACE EAS WITH INEAS, BIRTHDATE WITH INBRTH
REPLACE COMMENT WITH INCOMMNT
REPLACE FEIGHT WITH INHGHT, WEIGHT WITH INWEIGHT
REPLACE WTCNT WITH INWTCNT, GASMASK WITH !(INGAS)

USE B:EST INDEX B:ESTSSN
APPEND BLANK
REPLACE SSN WITH INSSN, COC WITH !(INCODE)
REPLACE HIS WITH !(INHIS), COD WITH !(INDRILL)
REPLACE INT WITH !(ININT), AID WITH !(INAID)
REPLACE UNI WITH !(INUNI)
REPLACE PFT1RAW WITH INPFT1, PFT1DATE WITH INPFTDT1
REPLACE PFT2RAW WITH INPFT2, PFT2DATE WITH INPFTDT2
REPLACE NBC WITH !(INNBC)
REPLACE MKS WITH !(INMKS), TAC WITH !(INTAC)

USE B:QUAL INDEX B:QUALSSN
APPEND BLANK

```



```

REPLACE SSN WITH INSSN, SWIMQUAL WITH !(INSWOU)
REPLACE SWIMDATE WITH INSWDT, EFDATE WITH INRFLDT
REPLACE EFSCORE WITH INRFLSC
REPLACE ESTSCORE WITH INPSTSC
REPLACE ESTDATE WITH INPSTDT

USE E:INFOMISC INDEX E:INFOSSN
APPEND BLANK
REPLACE SSN WITH INSSN, ELECWAR WITH !(INELEC)
REPLACE CLDWTHR WITH !(INCLDWTR)
REPLACE LAWWAR WITH !(INLAW), MOS WITH !(INMOS)
REPLACE DRUG WITH !(INDRUG), LDRSHP WITH INLDR
REPLACE ALCCOL WITH !(INALCH), HUMREL WITH !(INHUM)
REPLACE PERSAFFR WITE !(INPERSAF), UCMJ WITH !(INUCMJ)
REPLACE CHRMORED WITH !(INCHRMCR)
STORE 1 TO CALCOPT
DO WHILE CALCOPT < 6
    DO CALCULAT
    STCRE (CALCOPT + 1) TO CALCCPT
ENDDC

RELEASE ALL EXCEPT AA??????
ERASE
STORE AASSN TO VSSN
LO ITFSCFN
DO GEIDATA

STORE F TO INDONE
DO WHILE .NCT. INDONE
    ACCEPT "IS THIS DATA CORRECT (Y/N)?" TO INAGAIN
    DC CASE
        CASE !(INAGAIN) = 'Y'
            STCRE T TO INDCNE
        CASE !(INAGAIN) = 'N'
            DO UPDATA
            STCRE T TO INDONE
    ENDCASE
ENDDC
RELEASE ALL EXCEPT AA??????
RETURN

```


J. DATA ERROR CHECKING ROUTINE

```
* Routine Name: Error.prg
* Module Name: Maintenance Module
* Version: 6.3.1.1
* Author: D.P. Haeusler
* Date: 2 Dec 83
* Variables Used: aadate, in*,
* Variables Modified: in*
* Variables Created: inerror
* Variables Released: none
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Creatitr.prg, updata.prg
* Description: This routine does the error checking for the
* information entered in each of the appropriate fields.
* This error checking is not very extensive because this is
* a prototype system but can be increased if the need
* arises.
```

```
STORE AADATE TO INDATE
STORE T TO INERROR
DO WHILE INERROR
  STORE F TO INERROR
  DO CASE
    CASE ERROPT = 1
      IF .NOT. ($ (INRANK,1,1) = 'E'.OR. $ (INRANK,1,1) ;
        = W .OR. $ (INRANK,1,1) = 'O')
        STORE T TO INERROR
        @ 24,0 SAY " "
        @ 24,1 SAY "IMPROPER GRADE -- REENTER"
        @ 1,51 GET INRANK PICTURE "X9"
        READ
      ENDIF
    CASE ERROPT = 2
      IF .NOT. (VAL($ (INBIRTH,1,2)) < 31 .AND.
        VAL($ (INBIRTH,3,2)) < 13 .AND.
        VAL($ (INBIRTH,5,2)) < VAL($ (INDATE,5,2)))
        STORE T TO INERROR
        @ 24,0 SAY " "
        @ 24,1 SAY "IMPROPER DATE -- REENTER"
        @ 2,22 GET INBIRTH PICTURE "999999"
        READ
      ENDIF
    CASE ERROPT = 3
      IF .NOT. (VAL($ (INJNDATE,1,2)) < 31 .AND.;
        VAL($ (INJNDATE,3,2)) < 13 .AND.
        VAL($ (INJNDATE,5,2)) <= VAL($ (INDATE,5,2)))
        STORE T TO INERROR
        @ 24,0 SAY " "
        @ 24,1 SAY "IMPROPER DATE -- REENTER"
        @ 3,37 GET INJNDATE PICTURE "999999"
        READ
      ENDIF
    CASE ERROPT = 4
      IF .NOT. (! (INGAS) = 'S' .OR. ! (INGAS) = 'M' .OR.;
        ! (INGAS) = 'L')
        STORE T TO INERROR
        @ 24,0 SAY " "
        @ 24,1 SAY "IMPROPER GAS MASK SIZE -- REENTER"
        @ 3,68 GET INGAS PICTURE "X"
        READ
      ENDIF
    CASE ERROPT = 5
      IF .NOT. (INFT1 < 301)
        STORE T TO INERROR
        @ 24,0 SAY " "
        @ 24,1 SAY "IMPROPER PFT SCORE -- REENTER"
```



```

      @ 4,50 GET INPFT1 PICTURE "999"
      READ
    ENDIF
  CASE ERROPT = 6
    IF .NOT. (VAL($ (INPFTDT1,1,2)) < 31 .AND.;
      VAL($ (INPFTDT1,3,2)) < 13 .AND.;
      VAL($ (INPFTDT1,5,2)) = VAL($ (INDATE,5,2)) .CR.;
      INPFTDT1 = 'DDMMYY')
      STORE T TC INERROR
      @ 24,0 SAY "
      @ 24,1 SAY " IMPROPER PFT DATE -- REENTER"
      @ 4,65 GET INPFTDT1 PICTURE "999999"
      READ
    ENDIF
  CASE ERROPT = 7
    IF .NOT. (INPFT2 < 301)
      STORE T TC INERROR
      @ 24,0 SAY "
      @ 24,1 SAY " IMPROPER PFT SCORE -- REENTER"
      @ 5,50 GET INPFT2 PICTURE "999"
      READ
    ENDIF
  CASE ERROPT = 8
    IF .NOT. (VAL($ (INPFTDT2,1,2)) < 31 .AND.;
      VAL($ (INPFTDT2,3,2)) < 13 .AND.;
      VAL($ (INPFTDT2,5,2)) = VAL($ (INDATE,5,2)) .CR.;
      INPFTDT2 = 'DDMMYY')
      STORE T TC INERROR
      @ 24,0 SAY "
      @ 24,1 SAY " IMPROPER PFT DATE -- REENTER"
      @ 5,65 GET INPFTDT2 PICTURE "999999"
      READ
    ENDIF
  CASE ERROPT = 9
    IF .NOT. ($ (INSWQU,1,1) = 'S' .OR. $ (INSWQU,1,1);
      = 'S' .OR. INSWQU = 'WQ' .OR. INSWQU = 'wq')
      STORE T TC INERROR
      @ 24,0 SAY "
      @ 24,1 SAY " IMPROPER ENTRY -- REENTER"
      @ 11,19 GET INSWQU PICTURE "XX"
      READ
    ENDIF
  CASE ERROPT = 10
    IF .NOT. (VAL($ (INSWDT,1,2)) < 31 .AND.;
      VAL($ (INSWDT,3,2)) < 13 .AND.;
      VAL($ (INSWDT,5,2)) <= VAL($ (INDATE,5,2)) .CR.;
      INSWDT = 'DDMMYY')
      STORE T TC INERROR
      @ 24,0 SAY "
      @ 24,1 SAY " IMPROPER DATE -- REENTER"
      @ 11,28 GET INSWDT PICTURE "999999"
      READ
    ENDIF
  CASE ERROPT = 11
    IF .NOT. (INRFLSC < '251')
      STORE T TC INERROR
      @ 24,0 SAY "
      @ 24,1 SAY " IMPROPER RIFLE SCORE --- REENTER"
      @ 12,20 GET INRFLSC PICTURE "999"
      READ
    ENDIF
  CASE ERROPT = 12
    IF .NOT. (VAL($ (INRFLDT,1,2)) < 31 .AND.;
      VAL($ (INRFLDT,3,2)) < 13 .AND.;
      VAL($ (INRFLDT,5,2)) <= VAL($ (INDATE,5,2)) .CR.;
      INRFLDT = 'DDMMYY')
      STORE T TC INERROR
      @ 24,0 SAY "
      @ 24,1 SAY " IMPROPER DATE -- REENTER"

```



```

        @ 12,35 GET INRFLDT PICTURE "999999"
        READ
    ENDIF
CASE ERROPT = 13
    IF INWEIGHT > '300' .CR. INWEIGHT < '070'
        STORE T TC INERROR
        @ 24,0 SAY "
        @ 24,1 SAY " IMPROPER WEIGHT -- REENTER"
        @ 12,58 GET INWEIGHT PICTURE "999"
        READ
    ENDIF
CASE ERROPT = 14
    IF .NOT. (INESTSC < '301')
        STORE T TC INERROR
        @ 24,0 SAY "
        @ 24,1 SAY " IMPROPER PISTOL SCORE -- REENTER"
        @ 13,20 GET INPSTSC PICTURE "999"
        READ
    ENDIF
CASE ERROPT = 15
    IF .NOT. (VAL ($ (INPSTDT,1,2)) < 31 .AND.;
        VAL ($ (INPSTDT,3,2)) < 13 .AND.;
        VAL ($ (INPSTDT,5,2)) <= VAL ($ (INDATE,5,2)) .CR.;
        INPSTDT = 'DDMMYY' )
        STORE T TC INERROR
        @ 24,0 SAY "
        @ 24,1 SAY " IMPROPER DATE -- REENTER"
        @ 13,35 GET INPSTDT PICTURE "999999"
        READ
    ENDIF
CASE ERROPT = 16
    IF .NOT. (INHGHT < '78' .AND. INHGHT > '64')
        STORE T TC INERROR
        @ 24,0 SAY "
        @ 24,1 SAY " IMPROPER HEIGHT -- REENTER"
        @ 13,58 GET INHGHT PICTURE "99"
        READ
    ENDIF
ENICASE
ENDDC
RETURN

```


K. CALCULATION ROUTINE

```
* Routine Name: Calculat.prg
* Module Name: Maintenance Module
* Version: 6.3.1.2
* Author: D.P. Haeusler
* Date: 2 Dec 83
* Variables Used: calcopt, aassn, aadate, brthdate, age,
*                 mm, in*
* Variables Modified: age, mm, in*
* Variables Created: age, mm, in*
* Variables Released: all except aa??????
* Files (c/c): a:pers(o/c), b:qual(o/c), b:est(o/c)
* Temporary Files Created: none
* Using Subroutines: Updata.prg, indata.prg
* Description: This routine does the calculations that are
* triggered by the entry of data into certain fields of the
* ITR. These calculations reconstruct the tables found in
* the training orders.
```

```
DO CASE
CASE CALCOPT = 1
  USE A:PERS INDEX A:PERSSN
  FIND &AASSN
  IF INPFT1 > 0
    STCRE (VAL($ (AADATE,5,2))) -;
    (VAL($ (BRTHDATE,5,2))) TO AGE
    STCRE (VAL($ (AADATE,3,2))) -;
    (VAL($ (BRTHDATE,3,2))) TO MM
  IF MM < 0
    STORE AGE - 1 TO AGE
  ENDIF
  USE B:EST INDEX B:ESTSSN
  FIND &AASSN
  IF PFT1RAW >= 285
    STORE 'S' TO INPFT1CL
    REPLACE PFT1CLSS WITH INPFT1CL
    @ 4,57 SAY INPFT1CL
    RELEASE INPFT1CL
    USE
  ELSE
    IF AGE <= 26
      STORE PFT1RAW TO IPFT1RAW
    ENDIF
    IF AGE < 40 .AND. AGE > 26
      STORE (PFT1RAW + 25) TO IPFT1RAW
    ENDIF
    IF AGE > 39
      STORE (PFT1RAW + 50) TO IPFT1RAW
    ENDIF
    IF IPFT1RAW <= 134
      STORE 'U' TO INPFT1CL
      REPLACE PFT1CLSS WITH INPFT1CL
      @ 4,57 SAY INPFT1CL
      RELEASE INPFT1CL
      USE
    ENDIF
    IF IPFT1RAW <= 174 .AND. IPFT1RAW > 134
      STORE '3' TO INPFT1CL
      REPLACE PFT1CLSS WITH INPFT1CL
      @ 4,57 SAY INPFT1CL
      RELEASE INPFT1CL
      USE
    ENDIF
    IF IPFT1RAW <= 224 .AND. IPFT1RAW > 174
      STORE '2' TO INPFT1CL
      REPLACE PFT1CLSS WITH INPFT1CL
      @ 4,57 SAY INPFT1CL
```



```

        RELEASE INPFT1CL
        USE
    ENDIF
    IF IPFT1RAW > 224
        STORE '1' TO INEFT1CL
        REPLACE PFT1CLSS WITH INPFT1CL
        @ 4,57 SAY INPFT1CL
        RELEASE INPFT1CL
        USE
    ENDIF
ENDIF
ENDIF
CASE CALCOPT = 2
    USE A: PERS INDEX A: PERSSN
    FIND &AASSN
    IF INEFT2 > 0
        STCRE (VAL($ (AADATE,5,2))) -;
        VAL($ (BRTHDATE,5,2))) TO AGE
        STCRE (VAL($ (AADATE,3,2))) -;
        VAL($ (BRTHDATE,3,2))) TO MM
        IF MM < 0
            STORE AGE - 1 TC AGE
        ENDIF
        USE B: EST INDEX B: ESTSSN
        FIND &AASSN
        IF PFT2RAW >= 285
            STORE 'S' TO INPFT2CL
            REPLACE PFT2CLSS WITH INPFT2CL
            @ 5,57 SAY INPFT2CL
            RELEASE INPFT2CL
            USE
        ELSE
            IF AGE <= 26
                STORE PFT2RAW TC IPFT2RAW
            ENDIF
            IF AGE < 40 .AND. AGE > 26
                STORE (PFT2RAW + 25) TO IPFT2RAW
            ENDIF
            IF AGE > 39
                STORE (PFT2RAW + 50) TO IPFT2RAW
            ENDIF
            IF IPFT2RAW <= 134
                STORE 'U' TO INPFT2CL
                REPLACE PFT2CLSS WITH INPFT2CL
                @ 5,57 SAY INPFT2CL
                RELEASE INPFT2CL
                USE
            ENDIF
            IF IPFT2RAW <= 174 .AND. IPFT2RAW > 134
                STORE '3' TO INEFT2CL
                REPLACE PFT2CLSS WITH INPFT2CL
                @ 5,57 SAY INPFT2CL
                RELEASE INPFT2CL
                USE
            ENDIF
            IF IPFT2RAW <= 224 .AND. IPFT2RAW > 174
                STORE '2' TO INEFT2CL
                REPLACE PFT2CLSS WITH INPFT2CL
                @ 5,57 SAY INPFT2CL
                RELEASE INPFT2CL
                USE
            ENDIF
            IF IPFT2RAW > 224
                STORE '1' TO INEFT2CL
                REPLACE PFT2CLSS WITH INPFT2CL
                @ 5,57 SAY INPFT2CL
                RELEASE INPFT2CL
                USE
            ENDIF
        ENDIF
    ENDIF

```



```

ENDIF
ENCIF
CASE CALCOPT = 3
USE E:QUAL INDEX B:QUALSSN
FIND &AASSN
IF RFLSCORE > '0'
IF RFLSCORE < '190'
STORE 'UN' TO INRFLQU
REPLACE RFLQUAL WITH INRFLQU
@ 12,27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
IF RFLSCORE < '210' .AND. RFLSCORE >= '190'
STORE 'MM' TO INRFLQU
REPLACE RFLQUAL WITH INRFLQU
@ 12,27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
IF RFLSCORE < '220' .AND. RFLSCORE >= '210'
STORE 'SS' TO INRFLQU
REPLACE RFLQUAL WITH INRFLQU
@ 12,27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
IF RFLSCORE >= '220'
STORE 'EX' TO INRFLQU
REPLACE RFLQUAL WITH INRFLQU
@ 12,27 SAY INRFLQU
RELEASE INRFLQU
ENDIF
USE
ENCIF
CASE CALCOPT = 4
USE E:QUAL INDEX B:QUALSSN
FIND &AASSN
IF PSTSCORE > '0'
IF PSTSCORE < '180'
STORE 'UN' TO INPSTQU
REPLACE PSTQUAL WITH INPSTQU
@ 13,27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
IF PSTSCORE < '210' .AND. PSTSCORE >= '180'
STORE 'MM' TO INPSTQU
REPLACE PSTQUAL WITH INPSTQU
@ 13,27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
IF PSTSCORE < '250' .AND. PSTSCORE >= '210'
STORE 'SS' TO INPSTQU
REPLACE PSTQUAL WITH INPSTQU
@ 13,27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
IF PSTSCORE >= '250'
STORE 'EX' TO INPSTQU
REPLACE PSTQUAL WITH INPSTQU
@ 13,27 SAY INPSTQU
RELEASE INPSTQU
ENDIF
USE
ENCIF
CASE CALCOPT = 5
USE A:PERS INDEX A:PERSSN
FIND &AASSN
STORE T TO INWTCONT
IF HEIGHT = '64' .AND. WEIGHT <= '160' .AND.;
WEIGHT >= '105'
STORE F TO INWTCONT

```



```

        REPLACE WTCCNT WITH INWTCONT
        @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '65' .AND. WEIGHT <= '165' .AND.;
    WEIGHT >= '106';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '66' .AND. WEIGHT <= '170' .AND.;
    WEIGHT >= '107';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '67' .AND. WEIGHT <= '175' .AND.;
    WEIGHT >= '111';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '68' .AND. WEIGHT <= '181' .AND.;
    WEIGHT >= '115';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '69' .AND. WEIGHT <= '186' .AND.;
    WEIGHT >= '119';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '70' .AND. WEIGHT <= '192' .AND.;
    WEIGHT >= '123';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '71' .AND. WEIGHT <= '197' .AND.;
    WEIGHT >= '127';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '72' .AND. WEIGHT <= '203' .AND.;
    WEIGHT >= '131';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '73' .AND. WEIGHT <= '209' .AND.;
    WEIGHT >= '135';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '74' .AND. WEIGHT <= '214' .AND.;
    WEIGHT >= '139';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '75' .AND. WEIGHT <= '219' .AND.;
    WEIGHT >= '143';
    STORE F TO INWTCONT
    REPLACE WTCCNT WITH INWTCONT
    @ 11,59 SAY INWTCONT
ENDIF

```



```

IF HEIGHT = '76' .AND. WEIGHT <= '225' .AND.;
  WEIGHT >= '147';
  STCRE F TO INWTCONT
  REPLACE WTCCNT WITH INWTCONT
  @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '77' .AND. WEIGHT <= '230' .AND.;
  WEIGHT >= '151';
  STORE F TO INWTCONT
  REPLACE WTCCNT WITH INWTCONT
  @ 11,59 SAY INWTCONT
ENDIF
IF HEIGHT = '78' .AND. WEIGHT <= '235' .AND.;
  WEIGHT >= '153';
  STCRE F TO INWTCONT
  REPLACE WTCCNT WITH INWTCONT
  @ 11,59 SAY INWTCONT
ENDIF
IF INWTCONT
  REPLACE WTCCNT WITH INWTCONT
  @ 11,59 SAY INWTCONT
ENDIF
USE
ENDCASE
RETURN

```


1. COMMENT UPDATE ROUTINE

```

* Routine Name: Wrtcmmnt.prg
* Module Name: Maintenance Module
* Version: 6.3.1.3
* Author: D.P. Haeusler
* Date: 19 Jan 84
* Variables Used: aassn, intxt*
* Variables Modified: intxt*
* Variables Created: intxt*
* Variables Released: all like intxt*
* Files (opened/closed): a:cmmnt (o/c)
* Temporary Files Created: none
* Using Subroutines: indata.prg, updata.prg
* Description: This routine updates the information stored
* in the data base on an individual.

```

```

ERASE
@ 5,33 SAY "COMMENT SECTION"
USE 'A:CMNT INDEX A:CMNTSSN
FIND &AASSN
IF .NCT. # = 0
@ 8,2 SAY INFOTXT1
@ 8,51 SAY INFOTXT2
@ 9,2 SAY INFOTXT3
@ 9,51 SAY INFOTXT4
@ 10,2 SAY INFOTXT5
@ 10,51 SAY INFOTXT6
@ 11,2 SAY INFOTXT7
@ 11,51 SAY INFOTXT8
@ 12,2 SAY INFOTXT9
@ 12,51 SAY INFOTXT0
STORE INFOTXT1 TO INTXT1
STORE INFOTXT2 TO INTXT2
STORE INFOTXT3 TO INTXT3
STORE INFOTXT4 TO INTXT4
STORE INFOTXT5 TO INTXT5
STORE INFOTXT6 TO INTXT6
STORE INFOTXT7 TO INTXT7
STORE INFOTXT8 TO INTXT8
STORE INFOTXT9 TO INTXT9
STORE INFOTXT0 TO INTXT0
@ 8,1 GET INTXT1 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 8,50 GET INTXT2 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 9,1 GET INTXT3 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 9,50 GET INTXT4 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 10,1 GET INTXT5 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 10,50 GET INTXT6 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 11,1 GET INTXT7 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 11,50 GET INTXT8 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 12,1 GET INTXT9 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 12,50 GET INTXT0 PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
READ
REPLACE INFOTXT1 WITH INTXT1
REPLACE INFOTXT2 WITH INTXT2
REPLACE INFOTXT3 WITH INTXT3
REPLACE INFOTXT4 WITH INTXT4
REPLACE INFOTXT5 WITH INTXT5
REPLACE INFOTXT6 WITH INTXT6
REPLACE INFOTXT7 WITH INTXT7
REPLACE INFOTXT8 WITH INTXT8
REPLACE INFOTXT9 WITH INTXT9
REPLACE INFOTXT0 WITH INTXT0
ENDIF
USE
RELEASE ALL LIKE INTXT?
RETURN

```


M. CMMMENT CREATION ROUTINE

```
* Routine Name: Crtcmmnt.prg
* Module Name: Maintenance Module
* Version: 6.3.1.4
* Author: D.P. Haeusler
* Date: 10 Jan 84
* Variables Used: newtext, aassn
* Variables Modified: newtext
* Variables Created: newtext
* Variables Released: newtext
* Files (opened/closed): a:cmmnt (o/c)
* Temporary Files Created: none
* Using Subroutines: indata.prg, updata.prg
* Description: This routine creates a comment data base
* file for the individual that corresponds to the aassn.
* This file when created has 10 fields of 25 characters
* each and is filled with asterisks.
```

```
STORE "*****" TO NEWTXT
USE A:CMMNT INDEX A:CMMNTSSN
APPEND BLANK
REPLACE SSN WITH AASSN, INFCTXT1 WITH NEWTXT
REPLACE INFOTXT2 WITH NEWTXT, INFOTXT3 WITH NEWTXT
REPLACE INFCTXT5 WITH NEWTXT, INFOTXT6 WITH NEWTXT
REPLACE INFOTXT7 WITH NEWTXT, INFOTXT8 WITH NEWTXT
REPLACE INFOTXT9 WITH NEWTXT, INFOTXT0 WITH NEWTXT
REPLACE INFCTXT4 WITH NEWTXT
USE
RELEASE NEWTXT
RETURN
```


N. DELETE ITR ROUTINE

```
* Routine Name: Deletitr.prg
* Module Name: Maintenance Module
* Version: 6.4.0
* Author: D.P. Haeusler
* Date: 20 Jan 84
* Variables Used: del*, aassn
* Variables Modified: del*, aassn
* Variables Created: del*
* Variables Released: del*
* Files (opened/closed): a:pers(o/c), b:est(o/c),
                        b:qual(c/c), b:infomisc(o/c)
* Temporary Files Created: none
* Using Subroutines: Maintain.prg
* Description: This routine deletes records in the data
* base. It first checks to see if the record is in the
* data base, displays the record, queries the user if this
* is the record to be deleted, and deletes the record.
```

```
STORE F TO IELFINI
DO WHILE .NCT. DELFINI
  ERASE
  @ 12,30 SAY "ITR TO BE DELETED"
  ACCEPT "      NAME (LAST, FIRST MI.)====>" TO DELNAME
  USE A:PERS INDEX A:ALHPERS, A:PERSSN, A:MOSNDX
  FIND &DELNAME
  IF .NCT. # = 0
    STORE SSN TO AASSN
    DC ITFSCRN
    DO GETDATA
    STORE F TO DELTHRU
    DO WHILE .NOT. DELTHRU
      @ 23,0 SAY " IS THIS ITR TO BE DELETED? "
      ACCEPT " ENTER YES OR NO =====>" TO DELCHK
      IF !(DELCHK) = 'NO'
        STORE T TO DELTHRU
      ELSE
        IF !(DELCHK) = 'YES'
          ERASE
          @ 12,30 SAY "DELETING ITR"
          STORE T TO DELTHRU
          USE A:PERS INDEX A:ALHPERS, A:PERSSN,;
          A:MOSNDX
          FIND &AASSN
          IF .NCT. # = 0
            DELETE
            PACK
          ENDIF
          USE A:CMMNT INDEX A:CMMNTSSN
          FIND &AASSN
          IF .NCT. # = 0
            DELETE
            PACK
          ENDIF
          USE B:QUAL INDEX B:QUALSSN
          FIND &AASSN
          IF .NCT. # = 0
            DELETE
            PACK
          ENDIF
          USE B:EST INDEX B:ESTSSN
          FIND &AASSN
          IF .NCT. # = 0
            DELETE
            PACK
          ENDIF
          USE B:INFOMISC INDEX A:INFOSSN
```



```

        FIND &AASSN
        IF .NCT. # = 0
            DELETE
            PACK
        ENDIF
    ENDIF
ENDIF
ENDDO
ENLIF
STCRE F TO DEIDONE
DC WHILE .NOT. DEIDONE
    ERASE
    @ 12,20 SAY " DC YOU WANT TO DELETE ANOTHER RECORD? "
    ACCEPT "          ENTER YES OR NO =====> " TO DELCHK
    IF !(DELCHK) = 'NO'
        STCRE T TO DEIDONE
        STCRE T TO DELFINI
    ELSE
        IF !(DELCHK) = 'YES'
            STORE T TC DELDCNE
        ENDIF
    ENDIF
ENDIF
ENDDO
ENDDC
RELEASE ALL EXCEPT AA??????
RETURN

```


C. SYSTEM FUNCTION CONTROL ROUTINE

```
* Routine Name: System.prg
* Module Name: Maintenance Module
* Version: 6.5.0
* Author: D.P. Haeusler
* Date: 30 Jan 1984
* Variables Used: aacomple, sysopt
* Variables Modified: aacomple, sysopt
* Variables Created: aacomple, sysopt
* Variables Released: aacomple, sysopt
* Files (opened/closed): none
* Temporary Files Created: none
* Using Subroutines: Maintain.prg
* Description: This module creates the menu for all the
* system function modifications. It is a restricted module
* and can only be accessed by a user with a authorization
* level of 1. The module then can choose the appropriate
* subroutine corresponding to the option selected.
```

```
STORE F TO AACOMPLE
DO WHILE .NOT. AACOMPLE
  ERASE
  @ 6,29 SAY " SYSTEM FUNCTION MENU "
  @ 8,24 SAY " SELECT ONE OF THE FOLLOWING OPTIONS"
  @ 10,31 SAY "I...INSTALL SYSTEM"
  @ 11,31 SAY "S...SYSTEM RESET (YEARLY) "
  @ 12,31 SAY "A...ACCESS LIST MAINTENANCE"
  @ 13,31 SAY "C...QUIT TO MAINTENCE MENU"
  ACCEPT " ENTER OPTION ==>" TO SYSOPT
  DO CASE
    CASE !(SYSOPT) = "I"
      STORE F TO AACOMPLE
      DO B:INSTALL
    CASE !(SYSOPT) = "S"
      STORE F TO AACOMPLE
      DO B:SYSRESET
    CASE !(SYSOPT) = "A"
      STORE F TO AACOMPLE
      DO B:RSTACCSS
    CASE !(SYSOPT) = "Q"
      STORE T TO AACOMPLE
  ENDCASE
ENDDC
RELEASE ALL EXCEPT AA?????
RELEASE AACOMPLE
RETURN
```


F. SYSTEM INSTALLATION ROUTINE

```
* Routine Name: Install.prg
* Module Name: Maintenance Module
* Version: 6.5.1
* Author: D.P. Haeusler
* Date: 7 Feb 84
* Variables Used: *cc, *plt, crntyr
* Variables Modified: *co, *plt, crntyr
* Variables Created: none
* Variables Released: all
* Files (opened/closed): a:unitmem.mem, a:memdisk.mem
* Temporary Files Created: temp.mem
* Using Subroutines: system.prg
* Description: This routine stores the appropriate values
* in the memory files that the system uses to describe the
* units that are in the battalion. It also establishes
* the current year that is used for error checking and
* calculations
```

```
SAVE TO TEMP
RELEASE ALL
ERASE
RESTORE FROM A:UNITMEM
@ 3,30 SAY "SYSTEM INSTALLATION"
@ 5,15 SAY "ENTER THE NAMES OF YOUR COMPANY SIZE UNITS"
@ 7,15 SAY "COMPANY/EATTERY #1"
@ 7,34 GET CNECO PICTURE "XXXXXX"
@ 8,15 SAY "COMPANY/EATTERY #2"
@ 8,34 GET TWOCO PICTURE "XXXXXX"
@ 9,15 SAY "COMPANY/EATTERY #3"
@ 9,34 GET THREECO PICTURE "XXXXXX"
@ 10,15 SAY "COMPANY/EATTERY #4"
@ 10,34 GET FOURCO PICTURE "XXXXXX"
READ
STORE ! (CNECO) TO ONECO
STORE ! (TWOCO) TO TWOCO
STORE ! (THREECO) TO THREECO
STORE ! (FOURCO) TO FOURCO
ERASE
@ 3,30 SAY "SYSTEM INSTALLATION"
@ 5,15 SAY "ENTER THE NAMES OF PLATOON SIZED UNITS"
@ 7,15 SAY "PLATCON #1"
@ 7,26 GET CNEPLT PICTURE "XXXXXXXXXX"
@ 8,15 SAY "PLATCON #2"
@ 8,26 GET TWOPLT PICTURE "XXXXXXXXXX"
@ 9,15 SAY "PLATCON #3"
@ 9,26 GET THREEPLT PICTURE "XXXXXXXXXX"
@ 10,15 SAY "PLATOON #4"
@ 10,26 GET FOURPLT PICTURE "XXXXXXXXXX"
READ
STORE ! (CNEPLT) TO ONEPLT
STORE ! (TWOPLT) TO TWOPLT
STORE ! (THREEPLT) TO THREEPLT
STORE ! (FOURPLT) TO FOURPLT
SAVE TO UNITMEM
ERASE
RELEASE ALL
RESTORE FROM A:MEMDISK
@ 12,25 SAY "ENTER THE CURRENT YEAR"
ACCEPT " YEAR (YY) ==>" TO CRNTYR
SAVE TO A:MEMDISK
RELEASE ALL
RESTORE FROM TEMP
RETURN
```


Q. SYSTEM RESET ROUTINE

```
* Routine Name: Sysreset.prq
* Module Name: Maintenance Module
* Version: 6.5.2
* Author: L.P. Haeusler
* Date: 4 Feb 84
* Variables Used: crntyr, reset, rgrreset, done, q*, *co
* Variables Modified: crntyr, reset, rgrreset, done, q*, *co
* Variables Created: reset, rgrreset, done
* Variables Released: all
* Files (opened/closed): a:memdisk.mem, a:unitmem.mem,
* Temporary Files Created: temp.mem
* Using Subroutines: System.prq
* Description: This routine sets all the values to the
* default value for the fields that are reset yearly. The
* information is then stored in the data base for each
* individual. It also stores the yearly projections on
* the range quotas to a memory file that can be recalled by
* routines that use these values for calculations.
```

```
SAVE TO TEMP
RELEASE ALL
RESTORE FROM A:MEMDISK ADDITIVE
ERASE
@ 10,15 SAY "ENTER THE CURRENT YEAR"
ACCEPT " YEAR (YY) ==>" TO CRNTYR
SAVE TO A:MEMDISK
RELEASE ALL
STORE F TO DONE
DO WHILE .NOT. DONE
    ERASE
    @ 10,15 SAY "DO YOU WANT TO RESET RANGE QUOTAS FOR;
    THE YEAR?"
    ACCEPT " ENTER (Y OR N) ==>" TO RESET
    IF !(RESET) = 'Y'
        STORE T TO DONE
        STORE T TO RGRRESET
    ELSE
        IF !(RESET) = 'N'
            STORE T TO DONE
            STORE F TO RGRRESET
        ENDIF
    ENDIF
ENDDO
IF RGRRESET
    ERASE
    RESTORE FROM A:ARNGOTAS
    RESTORE FROM A:UNITMEM ADDITIVE
    @ 3,15 SAY "ENTER RANGE QUOTAS AS PROJECTED,;"
    FOR THE YEAR,
        @ 4,15 SAY "FOR UNIT: "
        @ 4,25 SAY CNECO
        @ 6,15 SAY "JANUARY"
        @ 6,23 GET QJAN PICTURE "999"
        @ 7,15 SAY "FEBRUARY"
        @ 7,24 GET QFEB PICTURE "999"
        @ 8,15 SAY "MARCH"
        @ 8,21 GET QMAR PICTURE "999"
        @ 9,15 SAY "APRIL"
        @ 9,21 GET QAPR PICTURE "999"
        @ 10,15 SAY "MAY"
        @ 10,19 GET QMAY PICTURE "999"
        @ 11,15 SAY "JUNE"
        @ 11,20 GET QJUN PICTURE "999"
        @ 12,15 SAY "JULY"
        @ 12,20 GET QJUL PICTURE "999"
        @ 13,15 SAY "AUGUST"
```



```

@ 13,22 GET QAUG PICTURE "999"
@ 14,15 SAY "SEPTEMBER"
@ 14,25 GET QSEP PICTURE "999"
@ 15,15 SAY "OCTOBER"
@ 15,23 GET QCCT PICTURE "999"
@ 16,15 SAY "NOVEMBER"
@ 16,24 GET QNOV PICTURE "999"
@ 17,15 SAY "DECEMBER"
@ 17,24 GET QDEC PICTURE "999"
READ
RELEASE ALL EXCEPT Q???
SAVE TO A:BRNGQTAS
RELEASE ALL
RESTORE FROM A:BRNGQTAS
RESTORE FROM UNITMEM ADDITIVE
ERASE
@ 3,15 SAY "ENTER RANGE QUOTAS AS PROJECTED,;
FOR THE YEAR,"
@ 4,15 SAY "FOR UNIT: "
@ 4,25 SAY TWCCO
@ 6,15 SAY "JANUARY"
@ 6,23 GET QJAN PICTURE "999"
@ 7,15 SAY "FEBRUARY"
@ 7,24 GET QFEB PICTURE "999"
@ 8,15 SAY "MARCH"
@ 8,21 GET QMAR PICTURE "999"
@ 9,15 SAY "APRIL"
@ 9,21 GET QAPR PICTURE "999"
@ 10,15 SAY "MAY"
@ 10,19 GET QMAY PICTURE "999"
@ 11,15 SAY "JUNE"
@ 11,20 GET QJUN PICTURE "999"
@ 12,15 SAY "JULY"
@ 12,20 GET QJUL PICTURE "999"
@ 13,15 SAY "AUGUST"
@ 13,22 GET QAUG PICTURE "999"
@ 14,15 SAY "SEPTEMBER"
@ 14,25 GET QSEP PICTURE "999"
@ 15,15 SAY "OCTOBER"
@ 15,23 GET QCCT PICTURE "999"
@ 16,15 SAY "NOVEMBER"
@ 16,24 GET QNOV PICTURE "999"
@ 17,15 SAY "DECEMBER"
@ 17,24 GET QDEC PICTURE "999"
READ
RELEASE ALL EXCEPT Q???
SAVE TO A:BRNGQTAS
RELEASE ALL
ERASE
RESTORE FROM A:BRNGQTAS
RESTORE FROM UNITMEM ADDITIVE
@ 3,15 SAY "ENTER RANGE QUOTAS AS PROJECTED,;
FOR THE YEAR,"
@ 4,15 SAY "FOR UNIT: "
@ 4,25 SAY THREECC
@ 6,15 SAY "JANUARY"
@ 6,23 GET QJAN PICTURE "999"
@ 7,15 SAY "FEBRUARY"
@ 7,24 GET QFEB PICTURE "999"
@ 8,15 SAY "MARCH"
@ 8,21 GET QMAR PICTURE "999"
@ 9,15 SAY "APRIL"
@ 9,21 GET QAPR PICTURE "999"
@ 10,15 SAY "MAY"
@ 10,19 GET QMAY PICTURE "999"
@ 11,15 SAY "JUNE"
@ 11,20 GET QJUN PICTURE "999"
@ 12,15 SAY "JULY"
@ 12,20 GET QJUL PICTURE "999"

```



```

@ 13,15 SAY "AUGUST"
@ 13,22 GET QAUG PICTURE "999"
@ 14,15 SAY "SEPTEMBER"
@ 14,25 GET QSEP PICTURE "999"
@ 15,15 SAY "OCTOBER"
@ 15,23 GET QCCT PICTURE "999"
@ 16,15 SAY "NOVEMBER"
@ 16,24 GET QNOV PICTURE "999"
@ 17,15 SAY "DECEMBER"
@ 17,24 GET QDEC PICTURE "999"
READ
RELEASE ALL EXCEPT Q???
SAVE TO A:CRNGQTAS
RELEASE ALL
ERASE
RESTORE FROM A:DRNGQTAS
RESTORE FROM UNITMEM ADDITIVE
@ 3,15 SAY "ENTER FANGE QUOTAS AS PROJECTED,;"
FOR THE YEAR, "
@ 4,15 SAY "FOR UNIT: "
@ 4,25 SAY FOURCO
@ 6,15 SAY "JANUARY"
@ 6,23 GET QJAN PICTURE "999"
@ 7,15 SAY "FEBRUARY"
@ 7,24 GET QFEB PICTURE "999"
@ 8,15 SAY "MARCH"
@ 8,21 GET QMAR PICTURE "999"
@ 9,15 SAY "APRIL"
@ 9,21 GET QAPR PICTURE "999"
@ 10,15 SAY "MAY"
@ 10,19 GET QMAY PICTURE "999"
@ 11,15 SAY "JUNE"
@ 11,20 GET QJUN PICTURE "999"
@ 12,15 SAY "JULY"
@ 12,20 GET QJUL PICTURE "999"
@ 13,15 SAY "AUGUST"
@ 13,22 GET QAUG PICTURE "999"
@ 14,15 SAY "SEPTEMBER"
@ 14,25 GET QSEP PICTURE "999"
@ 15,15 SAY "OCTOBER"
@ 15,23 GET QCCT PICTURE "999"
@ 16,15 SAY "NOVEMBER"
@ 16,24 GET QNOV PICTURE "999"
@ 17,15 SAY "DECEMBER"
@ 17,24 GET QDEC PICTURE "999"
READ
RELEASE ALL EXCEPT Q???
SAVE TO A:BRNGQTAS
RELEASE ALL
ENDIF
RESTORE FROM TEMP
STORE F TO DONE
DO WHILE .NOT. DONE
ERASE
@ 12,24 SAY "DO YOU WANT TO RESET YEARLY TRAINING;"
VALUES "
@ 13,24 SAY "IN THE DATA BASE FOR THE BATTALION?"
ACCEPT " ENTER (Y OR N) ==>" TO RESET
IF !(RESET) = 'Y'
STORE T TO DONE
USE E:EST
REPLACE ALL COC WITH "****"
REPLACE ALL HIS WITH "****"
REPLACE ALL COD WITH "****"
REPLACE ALL INT WITH "****"
REPLACE ALL AID WITH "*****"
REPLACE ALL UNI WITH "****"
REPLACE ALL PFT1RAW WITH 000
REPLACE ALL PFT1DATE WITH "DDMMYY"

```



```

REPLACE ALL PFT1CLSS WITH "*"
REPLACE ALL PFT2RAW WITH 000
REPLACE ALL PFT2DATE WITH "DDMMYY"
REPLACE ALL PFT2CLSS WITH "*"
REPLACE ALL NBC WITH "*****"
REPLACE ALL MKS WITH "*****"
REPLACE ALL TAC WITH "*****"
USE E:QUAL
REPLACE ALL RFIScore WITH "****"
REPLACE ALL RFIQUAL WITH "***"
REPLACE ALL RFIATE WITH "DDMMYY"
REPLACE ALL PSTIScore WITH "****"
REPLACE ALL PSTIQUAL WITH "***"
REPLACE ALL PSTIATE WITH "DDMMYY"
USE
ELSE
  IF !(RESET) = 'N'
    STORE T TO ICNE
  ENCIF
ENCIF
ENDDC
RELEASE ALL EXCEPT AA??????
RETURN

```


E. SYSTEM ACCESS CONTROL ROUTINE

```
* Routine Name: Rstaccss.prg
* Module Name: Maintenance Module
* Version: 6.5.3
* Author: D.P. Haeusler
* Date: 8 Feb 84
* Variables Used: thru, ac*, more, lines, done, cont
* Variables Modified: same as above
* Variables Created: same as above
* Variables Released: all
* Files (opened/closed): a:security(o/c)
* Temporary Files Created: none
* Using Subroutines: System.prg
* Description: This routine allows users with an access
* level of 1 to update the access list.
```

```
STORE F TO THRU
DO WHILE .NOT. THRU
  ERASE
  @ 8,28 SAY " SECURITY ACCESS LIST"
  @ 9,28 SAY " CHOOSE OPTION TO BE EXECUTED"
  @ 11,24 SAY " L...LIST ALL USERS"
  @ 12,24 SAY " A...ADD TO ACCESS LIST"
  @ 13,24 SAY " D...DELETE FROM ACCESS LIST"
  @ 14,24 SAY " Q...QUIT TC SYSTEM FUNCTION MENU"
  ACCEPT " ENTER OPTION ==>" TO ACCOPT
  DO CASE
    CASE !(ACCOPT) = "L"
      USE A:SECURITY INDEX A:ACCNAME
      DO WHILE .NOT. EOF
        ERASE
        @ 3,24 SAY " SECURITY ACCESS LIST"
        @ 5,1 SAY "NAME USER NO.;"
        PASSWORD LEVEL "
        STORE 7 TO LINES
        DO WHILE .NOT. EOF .AND. LINES < 22
          @ LINES,1 SAY NAME
          @ LINES,20 SAY USERID
          @ LINES,41 SAY USERPASS
          @ LINES,56 SAY AUTHLEV
          SKIP
          STORE (LINES + 2) TO LINES
          IF LINES > 22
            @ 23,15 SAY "*** PRESS ANY KEY TO;"
        CONTINUE ***"
        SET CONSOLE CFF
        WAIT
        SET CONSOLE CN
      ENDIF
    ENDDO
  ENDDO
  @ 23,15 SAY "*** PRESS ANY KEY TO CONTINUE ***"
  SET CONSOLE CFF
  WAIT
  SET CONSOLE CN
  USE
  CASE !(ACCOPT) = "A"
    STORE T TO MCRE
    DO WHILE MORE
      ERASE
      STORE " " TO ACNAME
      STORE " " TC ACCUSER
      STORE " " TO ACCPASS
      STORE 0 TO ACCLEV
      @ 10,25 SAY "NAME OF USER"
      @ 10,38 GET ACNAME PICTURE "XXXXXXXXXXXXXXXXXX"
      @ 12,25 SAY "USER ID"
```



```

@ 12,33 GET ACCUSER PICTURE "XXXXX"
@ 14,25 SAY "PASSWORD"
@ 14,34 GET ACCPASS PICTURE "XXXXXXXXXX"
@ 16,25 SAY "AUTHORIZATION LEVEL"
@ 16,45 GET ACCLEV PICTURE "9"
READ
STORE T TC NOGOOD
DO WHILE NOGOOD
    IF (ACCLEV < 1) .OR. (ACCLEV > 3)
        @ 23,1 SAY " ILLEGAL AUTHORIZATION LEVEL;
---REENTER"
        @ 16,45 GET ACCLEV PICTURE "9"
        READ
    ELSE
        STCFE F TC NCGOOD
    ENDIF
ENDDO
USE A:SECURITY INDEX A:ACCNAME, A:SECINDX
APPEND BLANK
REPLACE NAME WITH !(ACNAME) ,;
AUTHLEV WITH ACCLEV
REPLACE USERID WITH !(ACCUSER) ,;
USERPASS WITH !(ACCPASS)
USE
STORE F TC DONE
DO WHILE .NOT. DONE
    ERASE
    @ 10,25 SAY " DO YOU WANT TO ENTER ANCTHER;
USER?"
    ACCEPT "          ENTER (Y OR N ) ==>";
    TO CONT
    IF !(CCNT) = "Y"
        STCFE T TC DCNE
    ELSE
        IF !(CONT) = "N"
            STORE T TO DONE
            STORE F TC MORE
        ENDIF
    ENDIF
ENDDO
ENDDO
CASE !(ACCOPT) = "D"
    STORE T TO MCRE
    DO WHILE MORE
        ERASE
        @ 12,24 SAY " ENTER NAME OF THE USER TO BE;
DELETED"
        @ 13,24 SAY " OR QUIT TO REENTER ACCESS MENU "
        ACCEPT "          NAME ==>" TO ACNAME
        STORE !(ACNAME) TO ACNAME
        IF ACNAME = "QUIT"
            STORE F TC MCRE
        ELSE
            USE A:SECURITY INDEX A:ACCNAME, A:SECINDX
            FIND EACNAME
            IF # = 0
                @ 22,28 SAY "INDIVIDUAL NOT FOUND IN;
FILE"
                @ 23,24 SAY "*** PRESS ANY KEY TO;
CONTINUE ***"
                SET CONSOLE OFF
                WAIT
                SET CONSOLE ON
            ELSE
                DELETE
            ENDIF
        USE
    ENDIF
USE A:SECURITY INDEX A:ACCNAME, A:SECINDX

```



```
        PACK
        USE
      ENDDO
    CASE !(ACCOPT) = "Q"
      STCRE T TO THRU
    ENICASE
  ENDDC
RELEASE ALL EXCEPT AA??????
RETURN
```


218371

Thesis

P9464 Pruiett

c.1 An automated individual training record management systems (Prototype) United States Marine Corps.



thesP9464

An automated individual training record



3 2768 001 93042 3

DUDLEY KNOX LIBRARY